Tivoli Inventory

Benutzerhandbuch

Version 4.0

Tivoli Inventory

Benutzerhandbuch

Version 4.0

Tivoli Inventory Benutzerhandbuch

Copyrightvermerk

© Copyright IBM Corporation 2001. Alle Rechte vorbehalten. Kann nur gemäß der Softwarelizenzvereinbarung von Tivoli Systems, der Softwarelizenzvereinbarung von IBM oder dem Anhang für Tivoli-Produkte der IBM Nutzungsbedingungen oder Lizenzvereinbarungen verwendet werden. Diese Veröffentlichung darf ohne vorherige schriftliche Zustimmung der IBM Corporation weder ganz noch in Auszügen auf irgendeine Weise - elektronisch, mechanisch, magnetisch, optisch, chemisch, manuell u. a. - vervielfältigt, übertragen, aufgezeichnet, auf einem Abrufsystem gespeichert oder in eine andere Computersprache übersetzt werden.

Die IBM Corporation erteilt Ihnen eine eingeschränkte Berechtigung zur Erstellung einer Hardcopy oder sonstiger Vervielfältigungen in Form maschinenlesbarer Dokumentationen für die interne Verwendung, wobei jede Vervielfältigung den Copyrightvermerk der IBM Corporation enthalten muss. Weitere das Copyright betreffende Rechte werden nur nach vorheriger schriftlicher Zustimmung durch die IBM Corporation gewährt. Die Veröffentlichung dient nicht zu Produktionszwecken. **Die in diesem Dokument aufgeführten Beispiele sollen lediglich der Veranschaulichung und keinem anderen Zweck dienen.**

Marken

IBM, das IBM-Logo, Tivoli, das Tivoli-Logo, AIX, OS/2, Tivoli Enterprise, Tivoli Enterprise Console und TME sind in gewissen Ländern Marken oder eingetragene Marken der International Business Machines Corporation oder von Tivoli Systems Inc.

Microsoft, Windows, Windows NT und das Windows-Logo sind in gewissen Ländern Marken der Microsoft Corporation.

UNIX ist in gewissen Ländern eine eingetragene Marke von The Open Group.

Intel ist in gewissen Ländern eine Marke der Intel Corporation.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Bemerkungen

Hinweise auf IBM Produkte, Programme oder Dienstleistungen von Tivoli Systems oder IBM in dieser Veröffentlichung bedeuten nicht, dass diese in allen Ländern, in denen Tivoli Systems oder IBM vertreten ist, angeboten werden. Hinweise auf diese Produkte, Programme oder Dienstleistungen bedeuten nicht, dass nur Produkte, Programme oder Dienstleistungen von Tivoli Systems oder IBM verwendet werden können. Im Rahmen der gültigen gewerblichen oder anderen Schutzrechte von Tivoli Systems oder IBM können funktional äquivalente Produkte, Programme oder Dienstleistungen anstelle der angegebenen Produkte, Programme oder Dienstleistungen verwendet werden. Die Verantwortung für den Betrieb in Verbindung mit Fremdprodukten liegt beim Kunden, soweit solche Verbindungen nicht ausdrücklich von Tivoli Systems oder IBM bestätigt sind. Für die in diesem Dokument beschriebenen Produkte und Verfahren kann es Patente oder Patentanmeldungen von Tivoli Systems oder IBM geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France, zu richten. Anfragen an obige Adresse müssen auf Englisch formuliert werden.

Inhaltsverzeichnis

Vorwort xvii

Zielgruppe. xvii
Referenzliteratur xvii
Inhalt dieses Handbuchs xix
In diesem Handbuch verwendete Konventionen. xxi
Onlineveröffentlichungen. xxii
Veröffentlichungen bestellen xxii
Rückmeldungen zu Veröffentlichungen xxiii
Kundenunterstützung anfordern. xxiii

Kapitel 1. Einführung in Tivoli Inventory 1

Tivoli Inventory und SCS-Komponenten 2
Komponenten von Tivoli Inventory 3
SCS-Komponenten 4
Funktionsweise von Tivoli Inventory 6
Tivoli Inventory mit SCS (Scalable Collection Service) verwenden 8
Skalierbare Erfassung 8
Datenerfassung mit SCS 9
Tivoli Inventory- und SCS-Funktionen. 12
Tivoli Inventory-Funktionen. 12
Funktionen von SCS 15
Tivoli Inventory und andere Tivoli-Anwendungen 16
Sicherheit 17

Kapitel 2. Tivoli Inventory Version 4.0 installieren 19

Übersicht 19
Koexistenz mit früheren Versionen des Produkts 21

Tivoli Software Installation Service verwenden 21
SCS (Scalable Collection Service) installieren 22
SCS-Programmkorrektur installieren 25
Datenträger auswählen. 25
Tivoli Scalable Collection Service For Framework 3.7.1 installieren 27
Über die Befehlszeile 29
RIM und das Verwaltungssystem für relationale Datenbanken verwenden . . . 29
RIM-Host auswählen. 30
RDBMS-Server auswählen 31
Verwaltungssystem für relationale Datenbanken (RDBMS) konfigurieren . . . 31
DB2 konfigurieren. 32
Informix konfigurieren. 35
MS SQL-Server konfigurieren 35
Oracle konfigurieren 36
Sybase konfigurieren 37
Tivoli Inventory-Produkte auf Zielsystemen installieren - Vorbereitungen . . . 38
Datenträger auswählen. 39
Zwischen Tivoli Inventory Server oder Tivoli Inventory Gateway auswählen 43
Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren 43
Inventory über die Arbeitsoberfläche installieren 44
Kennwörter für RIM-Objekte ändern 51
Über die Befehlszeile 52
Tivoli Inventory Gateway installieren. 55
Tivoli Inventory deinstallieren 56
Tivoli Inventory mit dem automatischen Deinstallationsprogramm entfernen 57
Systeme aus dem Konfigurations-Repository entfernen. 60
TEC-Server angeben 60

Weitere Vorgehensweise. 61

Kapitel 3. Konfigurations-Repository erstellen 63

DB2-Datenbank als Konfigurations-Repository erstellen 64
Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen 65
Konfigurations-Repository-Schema installieren 67
Konfiguration prüfen 68
Weitere Vorgehensweise. 69
Informix-Datenbank als Konfigurations-Repository erstellen 69
Informix-Benutzer und Tivoli Inventory-Datenbank erstellen 70
Konfigurations-Repository-Schema installieren 71
Konfiguration prüfen 71
Weitere Vorgehensweise. 72
MS SQL-Server als Konfigurations-Repository erstellen 72
Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen 73
Konfigurations-Repository-Schema installieren 75
Konfiguration prüfen 76
Weitere Vorgehensweise. 76
Oracle-Datenbank als Konfigurations-Repository erstellen 77
Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen 77
Konfigurations-Repository-Schema installieren 79
Konfiguration prüfen 80
Weitere Vorgehensweise. 81
Sybase-Datenbank als Konfigurations-Repository erstellen 81
Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen 82
Konfigurations-Repository-Schema installieren 84
Konfiguration prüfen 84
Weitere Vorgehensweise. 85
Softwaresignaturen installieren 85

Tivoli Inventory- und Subskriptionsabfragen installieren 87
Weitere Vorgehensweise 90

Kapitel 4. Mit SCS arbeiten 91

SCS zusammen mit Tivoli Inventory verwenden 93
Kollektoren verwenden 95
Kollektoren stoppen, starten und zurücksetzen 95
Kollektoren konfigurieren 96
Das Laufzeitverzeichnis und das Depot konfigurieren 96
Die Protokolldatei für Kollektoren konfigurieren 98
Threads konfigurieren 98
Den Inventory Data Handler verwenden 99
Den Inventory Data Handler konfigurieren 100
Den Inventory Data Handler verschieben 102
Das Callback-Objekt von Inventory verschieben 103
RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren 105
RIM-Objekte erstellen 105
RIM-Objekte konfigurieren 107
Datenerfassung terminieren 109
Datenfluss im Netzwerk steuern 112
Informationen zu SCS abrufen 114
Kollektor-Konfiguration 115
Kollektorstatus 115
CTOCs und Warteschlangen 115
Inventory Data Handler-Konfiguration 116
Inventory-Suchläufe 117
Inventory-Profile 117
Im Konfigurations-Repository gespeicherte Daten 117
Statusinformationen in Tivoli Inventory konfigurieren 117

Hinweise zu miteinander verbundenen TMRs 119
Weitere Vorgehensweise 120

Kapitel 5. Benutzerspezifische Daten mit Tivoli Inventory erfassen 121

Mit Softwaresignaturen arbeiten 122
Über die Arbeitsoberfläche 124
Über die Befehlszeile 125
Benutzerdaten erfassen 126
Übersicht 127
Erfassung von Benutzerdaten vorbereiten 129
Benutzerzugriff auf UserLink erteilen 129
Über die Arbeitsoberfläche 131
Über die Befehlszeile 134
Tabellen für Benutzerdaten erstellen 135
Benutzerspezifische Tabellen, Anzeigen und Spalten für DB2 und Informix benennen 138
Protokolltabellen für benutzerspezifische Tabellen erstellen 140
Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen 141
MIF-Gruppen und -Attribute anhand der Benutzerdatenschablone erstellen 143
Datei userlink.htm verwenden 152
Benutzerdatenformular ausfüllen 153
Benutzerdaten abrufen und speichern 156
Benutzerdaten anzeigen 156
MIF-Benutzerdateien verwenden 158
PC-Suchläufe für die Erfassung von BIOS-Daten anpassen 161
Weitere Vorgehensweise 162

Kapitel 6. Mit Inventory-Profilen arbeiten 163

Inventory-Profile verwenden 164
Inventory-Profile erstellen 165
Über die Arbeitsoberfläche 167
Über die Befehlszeile 171
Inventory-Profile anpassen 171
Allgemeine Merkmale anpassen 172
Softwaresuchläufe anpassen 174
Optionen für PC-Softwaresuchlauf 174
Optionen für UNIX-Softwaresuchlauf 177
Optionen für die Konfiguration des Softwaresuchlaufs 178
Suchlaufoptionen 179
Dateien 182
Verzeichnisse 183
Hardwaresuchläufe anpassen 183
PC-Hardwaresuchläufe 183
Tivoli Hardware-Suchfunktion 184
DMI-Suchfunktion 185
UNIX-Hardwaresuchläufe 188
Benutzerdefinierte Skripts und MIF-Dateien verwenden 189
Berechtigungsklassen 191
Über die Arbeitsoberfläche 191
Über die Befehlszeile 191
Inventory-Profile klonen 192
Über die Arbeitsoberfläche 192
Über die Befehlszeile 193
Inventory-Profile löschen 194
Über die Arbeitsoberfläche 195
Über die Befehlszeile 195
Inventory-Profile umbenennen 196

Weitere Vorgehensweise 196

Kapitel 7. Inventory-Profile verteilen 197

Inventory-Profile verteilen 198
Über die Arbeitsoberfläche 200
Über die Befehlszeile 204
Von Endpunkten initiierten Suchlauf ausführen 204
Daten im Konfigurations-Repository speichern 206
Weitere Vorgehensweise 207

Kapitel 8. Inventory-Daten abfragen 209

Inventory- und Subskriptionsabfragen verwenden 210
Abfragen erstellen 211
Über die Arbeitsoberfläche 212
Über die Befehlszeile 220
Abfragen ausführen 221
Über die Arbeitsoberfläche 222
Über die Befehlszeile 224
Subskribenten über Abfragen ermitteln 225
Über die Arbeitsoberfläche 226
Befehlszeile 228
Informationen zu einzelnen Clients abrufen 229
Über die Arbeitsoberfläche 230
Über die Befehlszeile 232
Abfragen im TMR-Verbund verwenden 233

Anhang A. Berechtigungsklassen 235

Inventory-Berechtigungsklassen 235
Inventory-Operationen 236

Abfrageoperationen 237

Anhang B. Befehle 239

Tivoli-Befehle verwenden 239

Befehlszeilensyntax 240

Beispiel 240

Objektverweise 241

Registrierte Namen 241

Objektpfade 242

Beispiele 243

Befehle für Tivoli Inventory 243

Befehle für SCS 245

Befehlssyntax 246

wcollect 247

wcrtinvcb 256

wcrtinvdh 258

wcstat 262

wdistinv 265

wepscan 271

wgetinvdh 274

wgetinvglobal 277

wgetinvpcfiles 282

wgetinvpchw 285

wgetinvpcsw 288

wgetinvunixfiles 294

wgetinvunixhw 297

wgetinvunixsw 300

wgetscanstat 306
winvfilter 308
winvrmnode. 311
winvsig 313
wqueryinv 318
wsetinvdh 321
wsetinvglobal 325
wsetinvpcfiles 331
wsetinvpchw 337
wsetinvpcsw 342
wsetinvunixfiles. 348
wsetinvunixhw. 354
wsetinvunixsw. 358

Anhang C. Konfigurations-Repository 365

Protokollfunktion. 366
Protokolltabellen verwenden 367
Benutzerdefinierte Protokolltabellen 368
Protokolltabellen löschen 368
Ansichten im Konfigurations-Repository 370
CDROM_VIEW 370
COMPUTER_MEM_VIEW. 372
COMPUTER_VIEW 373
FLPY_DRV_VIEW 374
HDISK_VIEW 375
HEADER_INFO_VIEW 376
INST_FILE_VIEW 377
INST_SWARE_VIEW 379

INVENTORYDATA 380
IP_ADDR_VIEW 381
IPX_ADDR_VIEW 383
KEYBOARD_VIEW 384
MATCH_SWARE_VIEW 385
MOUSE_VIEW 386
NATIV_SWARE_VIEW 387
NET_CARD_VIEW 388
NOSIG_FILES_VIEW 389
NW_SERVER_VIEW 390
NW_VOLS_VIEW 393
OS_VIEW 394
PARTITION_VIEW 395
PC_BIOS_VIEW 397
PC_PROCESSOR_VIEW 398
PCI_DEV_VIEW 402
PRINTER_VIEW 403
PROCESSOR_VIEW 404
STORAGE_DEV_VIEW 405
SWARE_MATCH_CRC32 406
SWARE_MATCH_MD5 407
SWARE_MATCH_QUICK 409
USB_DEV_VIEW 410
VID_CARD_VIEW 411
Protokollansichten im Konfigurations-Repository 413
Tivoli Inventory- und Subskriptionsabfragen 415
Tivoli Inventory-Abfragen 415
Tivoli Inventory-Protokollabfragen 435
Subskriptionsabfragen 436
Protokollsubskriptionsabfragen 438

Tabellen im Konfigurations-Repository. 439
COMPUTER. 440
COMPUTER_SYS_MEM 441
FILE_DESC 441
FILE_FILTER. 442
FILE_PATH. 442
HDISK 443
HEADER_INFO 443
INST_HEADER_INFO 444
INST_MOUSE 444
INST_NATIV_SWARE 445
INST_PARTITION 445
INST_PRINTER 446
INST_PROCESSOR 446
INST_USB_DEV 447
INST_VID_CARD 447
IP_ADDR 448
IPX_ADDR. 449
MATCHED_SWARE. 449
MOUSE 449
NATIV_SWARE 450
NET_ADAPTER. 451
NW_SERVER. 451
NW_VOLS 452
PC_SYS_PARAMS 453
PCI_DEV 454
PRINTER 454
PROCESSOR 454
STORAGE_DEV. 456
SWARE_SIG. 457

UNIX_SYS_PARAMS. 457
UNMATCHED_FILES 458
USB_DEV 459
VID_CARD 459
Protokolltabellen 460

Anhang D. Richtlinien. 463

Standardwertegruppenmethoden 464
Richtlinienobjekte 465
Neue Richtlinienobjekte erstellen 466
Inhalt von Richtlinienmethoden ersetzen 467
Richtlinie einem Richtlinienbereich zuweisen 469
Standardwertegruppenmethode festlegen—Beispiel 470
Richtlinienmethoden 475
ic_def_global_action 476
ic_def_global_ep_timeout 477
ic_def_global_logfile_host 478
ic_def_global_logfile_path 479
ic_def_global_notice_interval 480
ic_def_global_notice_location 481
ic_def_global_notice_type 482
ic_def_global_update_replace 483
ic_def_pc_custom_mifs 484
ic_def_pc_custom_script 485
ic_def_pc_exclude_dirs 486
ic_def_pc_extensions 487
ic_def_pc_hw_gran 488
ic_def_pc_hw_outfile 490
ic_def_pc_hw_scan 491
ic_def_pc_include_dirs 492

ic_def_pc_sw_crc 493
ic_def_pc_sw_flags 494
ic_def_pc_sw_outfile 495
ic_def_pc_sw_scan 497
ic_def_unix_custom_mifs 498
ic_def_unix_custom_script 499
ic_def_unix_exclude_dirs 500
ic_def_unix_extensions 501
ic_def_unix_hw_gran 502
ic_def_unix_hw_outfile 503
ic_def_unix_hw_scan 504
ic_def_unix_include_dirs 505
ic_def_unix_sw_crc 506
ic_def_unix_sw_flags 507
ic_def_unix_sw_outfile 508
ic_def_unix_sw_scan 510

Anhang E. Fehlerbehebung 511

Konfigurations-Repository-Schema 511
DMI 511
DMI-Ebene hinzufügen 512
DMI-Suchlauf 514
DMI-Tabellen- und Spaltennamen 515
Endpunkt-initiierte Suchläufe 515
Grafische Benutzerschnittstelle 518
Architekturvoraussetzungen 518
Die Protokolldatei aktivieren 518
Die grafische Benutzerschnittstelle (GUI) aktivieren 519
Scalable Collection Service (SCS) 520

Protokoll- und Datendateien 520
Datendateien von Warteschlangen 521
Protokolldatei des Kollektors 522
Protokolldatei des Inventory Data Handler 523
Protokolldatei des Kollektor-Managers 523
Tivoli Inventory-Statusprotokolldatei 524
Statusprotokolldatei der beendeten CTOCs 525
Weitere Protokolldateien 525
Depot-Inhalt 527
Verfahren zur Fehlerbehebung 527
Abfragen 535

Glossar 537

Index 549

Vorwort

Das *Tivoli Inventory Benutzerhandbuch* enthält Informationen zur Verwaltung des Hardware- und Softwareinventars auf PCs und UNIX-Systemen. Tivoli Inventory verwendet Profile zum Suchen von Informationen und Tivoli Inventory Konfigurations-Repositorys zum Speichern von Informationen. Das vorliegende Handbuch enthält eine Einführung in das Produkt und Installationsanweisungen; darüber hinaus finden Sie hier Hinweise zu den Verwaltungs-Tasks, die mit Hilfe von Tivoli Inventory ausgeführt werden können, mit entsprechenden Verfahrenshinweisen und Beispielen.

Zielgruppe

Dieses Handbuch richtet sich an Systemadministratoren sowie alle Benutzer von Tivoli Inventory, die für die Erfassung von Bestandsdaten und Erstellung entsprechender Berichte zu PCs und UNIX-Systemen in verteilten Unternehmensumgebungen verantwortlich sind. Beim Benutzer werden Vorkenntnisse in den folgenden Bereichen vorausgesetzt:

- UNIX- und PC-Architekturen
- Grundprinzipien der Bestandskontrolle und der Systemkonfiguration
- Datenbank- und SQL-Konzepte
- Grafische Benutzerschnittstellen

Referenzliteratur

Vor der Installation von Tivoli Inventory sollten Sie die *Tivoli Inventory Release Notes Version 4.0* lesen. Die Release-Notes enthalten folgende Informationen:

- Neue Funktionen in Tivoli Inventory Version 4.0
- Systemvoraussetzungen, unter anderem unterstützte Plattformen, Web-Browser sowie der Platzbedarf für das Konfigurations-Repository

- Installationshinweise
- Fehler, die durch Tivoli Inventory Version 4.0 behoben wurden
- Bekannte Fehler und Möglichkeiten zu ihrer Vermeidung
- Produktinformationen zu Tivoli Inventory Version 4.0

Weitere Informationen zu den in Tivoli Inventory zur Verfügung gestellten Verwaltungsoptionen können Sie den folgenden Handbüchern entnehmen (dabei handelt es sich um die erforderliche Referenzliteratur):

- *Tivoli Management Framework: Einsatzplanung, Tivoli Enterprise Installation Guide, Tivoli Management Framework Benutzerhandbuch, Tivoli Management Framework Reference Manual* und *Tivoli Management Framework Maintenance and Troubleshooting Guide*.

  Enthält wichtige Informationen zur Konfiguration und Verwendung der Tivoli-Umgebung.

- *Tivoli Software Distribution Benutzerhandbuch*.

  Dieses Handbuch enthält Referenzinformationen zum Einsatz von Software mit Hilfe von Tivoli.

- *Tivoli Inventory Online-Hilfe*.

  Enthält Referenzinformationen zur Verwendung der grafischen Benutzerschnittstelle (GUI) von Tivoli Inventory.

Inhalt dieses Handbuchs

Das *Tivoli Inventory Benutzerhandbuch* ist in die folgenden Kapitel untergliedert:

- Einführung in Tivoli Inventory

  Dieses Kapitel enthält eine Übersicht über die Funktionsweise von Tivoli Inventory; darüber hinaus finden Sie hier eine Liste der verfügbaren Funktionen, eine Beschreibung des Zusammenspiels zwischen Tivoli Inventory und anderen Tivoli-Anwendungen sowie Informationen zur Konfiguration der Sicherheit für Tivoli Inventory-Tasks und -Daten.

- Tivoli Inventory Version 4.0 installieren

  In diesem Kapitel werden die einzelnen Schritte erläutert, die bei der Installation von Tivoli Inventory Version 4.0 in Ihrer Tivoli-Umgebung erforderlich sind.

- Konfigurations-Repository erstellen

  In diesem Kapitel werden die einzelnen Schritte erläutert, die bei der Erstellung der Schemas des Konfigurations-Repositorys von Tivoli Inventory, bei der Installation von Softwaresignaturen im Konfigurations-Repository sowie bei der Installation der Abfragen, über die Daten aus dem Konfigurations-Repository abgerufen werden, erforderlich sind.

- Konfigurations-Repository erstellen

  Dieses Kapitel enthält Informationen zur Konfiguration und Verwendung von SCS (SCS = Scalable Collection Service). SCS ist ein Tivoli Management Framework-Service, der eine effektive, asynchrone Erfassung einer großen Anzahl von Daten in mehreren komplexen Netzwerken ermöglicht.

- Mit Inventory-Profilen arbeiten

  In diesem Kapitel werden die einzelnen Schritte erläutert, die zum Erstellen, Anpassen, Klonen und Löschen von Inventory-Profilen erforderlich sind.

- Inventory-Profile verteilen

  In diesem Kapitel werden die einzelnen Schritte bei der Verteilung von Inventory-Profilen über die Tivoli-Arbeitsoberfläche und über UserLink erläutert. Darüber hinaus wird in diesem Kapitel erläutert, wie Suchvorgänge von Endpunkten aus gestartet werden.

- Benutzerspezifische Daten mit Tivoli Inventory erfassen

  Dieses Kapitel enthält Informationen zur Vorgehensweise bei der Erfassung benutzerspezifischer und optionaler Daten mit Hilfe von Tivoli Inventory. Sie haben die Möglichkeit, durch Hinzufügen von Softwaresignaturen im Konfigurations-Repository benutzerspezifische Daten zu erfassen. Darüber hinaus können Sie benutzerdefinierte Tabellen hinzufügen und mit Hilfe von UserLink benutzerdefinierte Daten erfassen, mit denen diese Tabellen aufgefüllt werden. In diesem Kapitel wird darüber hinaus auch erläutert, wie Benutzer Ihnen benutzerdefinierte Daten zur Verfügung stellen können.

- Inventory-Daten abfragen

  In diesem Kapitel werden die Vorgänge erläutert, die bei der Datensuche im Konfigurations-Repository erforderlich sind. Die Abfragefunktion von Tivoli Management Framework ermöglicht Ihnen die Erstellung von SQL-Abfragen, mit deren Hilfe Sie die Daten abfragen können, die über die Verteilung von Inventory-Profilen erfasst wurden.

- Berechtigungsklassen

  Dieser Anhang enthält Tabellen, in denen die verschiedenen Berechtigungsklassen erläutert werden, die für die Ausführung von Aufgaben mit Hilfe von Tivoli Inventory erforderlich sind.

- Befehle

  In diesem Anhang sind in alphabetischer Reihenfolge alle Tivoli-Befehle aufgeführt und erläutert, die für Tivoli Inventory und SCS relevant sind.

- Konfigurations-Repository

  In diesem Anhang sind alle Tabellen und Sichten aufgeführt und erläutert, die im Schema des Konfigurations-Repositorys von Tivoli Inventory Version 4.0 enthalten sind. Darüber hinaus sind hier alle Abfragen aufgeführt und erläutert, die bei der Installation von Inventory Version 4.0 installiert werden können.

- Richtlinien

  In diesem Anhang werden verschiedene Richtlinien erläutert; darüber hinaus wird hier auf die einzelnen Richtlinien eingegangen, die in Tivoli Inventory zur Anwendung kommen.

- Berechtigungsklassen

  Dieser Anhang enthält Informationen zur Fehlerbehebung in Tivoli Inventory.

- Glossar

  Das Glossar enthält Begriffe und Definitionen, die im Zusammenhang mit Tivoli Inventory von Bedeutung sind.

In diesem Handbuch verwendete Konventionen

Im vorliegenden Handbuch werden für bestimmte Begriffe und Aktionen verschiedene Schriftbilder verwendet. Diese Konventionen haben die folgende Bedeutung:

Fett Befehle, Schlüsselwörter, Berechtigungsklassen und andere unverändert zu übernehmende Informationen sind **fett** gedruckt.

Kursiv Vom Benutzer einzugebende Variablen und Werte und neue Begriffe sind *kursiv* gedruckt. Hervorhebungen von Wörtern und Ausdrücken im Text sind ebenfalls *kursiv*.

`Monospace-Schrift` Codebeispiele, Ausgaben und Systemnachrichten sind in der Schriftart `Monospace` gedruckt.

Onlineveröffentlichungen

Auf der Tivoli Customer Support-Website (http://www.tivoli.com/support/) finden Sie ein Handbuch zur Kundenunterstützung (das *Customer Support Handbuch*, nur in Englisch verfügbar), häufig gestellte Fragen (FAQs), technische Informationen, Release-Notes, Benutzerhandbücher, Redbooks und White Papers. Die Online-Veröffentlichungen finden Sie unter folgender Adresse:

http://www.tivoli.com/support/documents/.

Die Dokumentation für einige Produkte ist sowohl im PDF- als auch im HTLM-Format verfügbar. Ein Teil dieser Dokumentationen liegt in mehreren Sprache vor.

Für den Zugriff auf die Dokumentationen ist oftmals die Angabe einer ID und eines Kennworts erforderlich. Eine ID, die Sie für die Kundenunterstützungs-Website benötigen, erhalten Sie unter folgender Adresse: http://www.tivoli.com/support/getting/.

Personen, die das Produkt weiterveräußern, können der Website http://www.tivoli.com/support/smb/index.html weitere Informationen zum Erhalt technischer Dokumentationen und Unterstützung für Tivoli entnehmen.

Geschäftspartner finden unter “Veröffentlichungen bestellen” weitere Hinweise, wie Sie technische Dokumentationen für Tivoli beziehen können.

Veröffentlichungen bestellen

Tivoli-Veröffentlichungen können Sie per E-Mail (LITERATA@DE.IBM.COM), Telefon oder Fax unter den folgenden Nummern bestellen:

- Telefon: 0049-180-55090 bzw. 0180-55090
- Fax: 07032-15-3300

Rückmeldungen zu Veröffentlichungen

Bitte teilen Sie uns mit, welche Erfahrungen Sie mit Produkten und zugehörigen Dokumentationen von Tivoli gemacht haben. Verbesserungsvorschläge sind stets erwünscht. Falls Sie Anmerkungen bzw. Vorschläge zu unseren Produkten und Dokumentationen haben, können Sie uns wie folgt erreichen:

- Schicken Sie eine E-Mail an die folgende Adresse (bitte formulieren Sie die E-Mail in englischer Sprache): pubs@tivoli.com.
- Beteiligen Sie sich an unserer Kundenumfrage im Internet unter: http://www.tivoli.com/support/survey/.

Kundenunterstützung anfordern

Das Handbuch *Tivoli Customer Support Handbook* auf der Website http://www.tivoli.com/support/handbook/ enthält Informationen zu allen Aspekten der Tivoli-Kundenunterstützung einschließlich der folgenden Informationen:

- Registrierung und Anspruch.
- Anforderung von Kundenunterstützung entsprechend der Dringlichkeit Ihres Problems
- Telefonnummern und E-Mail-Adressen für Ihr Land
- Informationen, die Sie für die Kundenunterstützung bereithalten sollten.

1

Einführung in Tivoli Inventory

Die Erweiterung unternehmensweiter Netzwerke durch den Zusammenschluss räumlich zum Teil weit auseinanderliegender Standorte sowie der Einsatz von Systemen und Software verschiedener Hersteller führt zu einer zunehmend komplexeren Client-/Server-Verwaltung. Bei Tivoli Management Framework handelt es sich um eine verteilte, objektorientierte Struktur, die vor allem als Basis für den Einsatz sicherer, heterogener und skalierbarer Verwaltungsanwendungen konzipiert wurde. Diese Anwendungen, wie beispielsweise Tivoli Inventory, bieten Lösungen für die Verwaltung komplexer Client-/Server-Umgebungen. Tivoli Inventory verwendet unter anderem SCS (SCS = Scalable Collection Service). SCS hatte früher die Bezeichnung MCollect. SCS ist ein Tivoli Management Framework-Service, der eine effektive und asynchrone Erfassung großer Datenmengen in komplexen Netzwerken ermöglicht

Dieses Kapitel enthält eine Übersicht über die Funktionsweise von Tivoli Inventory und SCS; darüber hinaus finden Sie hier eine Liste der verfügbaren Funktionen, eine Beschreibung des Zusammenspiels zwischen Tivoli Inventory und SCS und anderen Tivoli-Anwendungen sowie Informationen zur Konfiguration der Sicherheit für Tivoli Inventory-Tasks und -Daten.

Bei den herkömmlichen Verfahren zur Erfassung des Inventars müssen Sie die Informationen zum Software- und Hardwareinventar an jeder Maschine erfassen und anschließend in ein Tabellenkalkulations- oder Datenbankprogramm eingeben.

Bei jeder Software- bzw. Hardwareaufrüstung durch den Benutzer müssen diese Daten in der Tabellenkalkulation bzw. Datenbank aktualisiert werden.

Dieses Verfahren ist zeitraubend, die ständige Aktualisierung der Daten gestaltet sich schwierig. Dies bedeutet, dass die Inventarinformationen von Administratoren und Mitarbeitern der Leistungsverrechnung nicht automatisch für System- und Softwareaktualisierungen oder verwaltungstechnische Aufgaben herangezogen werden können. Mit Tivoli Inventory dagegen ist eine schnelle, genaue und einfache Erfassung von Bestandsdaten in einer verteilten Umgebung und deren ständige Aktualisierung möglich.

Tivoli Inventory erfasst das Hardware- und Softwareinventar und unterstützt so Systemadministratoren und das Leistungsverrechnungspersonal bei der Verwaltung komplexer verteilter Client-/Server-Unternehmensumgebungen. Tivoli Inventory unterstützt Administratoren und das Leistungsverrechnungspersonal bei den folgenden Aufgaben:

- Zentrale Verwaltung aller unternehmensinternen Systeme
- Ermitteln der installierten Softwarebasis
- Bestätigung von Softwareverteilungen
- Ergänzen und Ersetzen von Inventurfunktionen
- Hilfe bei der Beschaffungsplanung
- Überprüfung von Softwarevoraussetzungen
- Steuerung der Vermögenswerte

Tivoli Inventory und SCS-Komponenten

In den folgenden Abschnitten werden die Komponenten beschrieben, die Sie benötigen, um Tivoli Inventory sowie SCS zur Datenerfassung in Tivoli Inventory zu verwenden.

Komponenten von Tivoli Inventory

Tivoli Inventory erfasst und speichert Informationen mit Hilfe der folgenden Komponenten:

- Dem TMR-Server (TMR= Tivoli Management Region), der Abfragen an die Zielsysteme absetzt.
- Der Tivoli Inventory-Anwendung, die auf jeden Fall auf dem TMR-Server installiert werden muss. Diese Anwendung enthält sowohl die grafische Benutzerschnittstelle (GUI) als auch die Befehlszeilenschnittstelle (CLI).
- Der Tivoli Inventory Gateway-Anwendung, die auf allen verwalteten Knoten installiert werden muss, die als Gateway für die Verteilung von Inventory-Profilen an Endpunkte eingesetzt werden sollen.
- Dem RDBMS-Server (RDBMS = Relational Database Management System) und dem *Konfigurations-Repository* von Tivoli Inventory im Verwaltungssystem für relationale Datenbanken. Bei dem Konfigurations-Repository handelt es sich um die RDBMS-Datenbank mit dem Schema (Tabellen und Spalten), in dem die Software- und Hardwareinventarinformationen gespeichert werden. Das Schema des Konfigurations-Repositorys stellt eine Struktur zum Speichern der Daten zur Verfügung, die bei Inventory-Abfragen erfasst werden.
- Dem RIM-Host *(RIM = RDBMS Interface Module)*, der die Kommunikation zwischen Tivoli Inventory und RDBMS-Datenbanken ermöglicht.
- *UserLink for Tivoli Inventory*, über das Benutzer Aufgaben im Zusammenhang mit Tivoli Inventory ausführen können, unter anderem das Ausfüllen des Benutzerdatenformulars.
- Der Web-basierten Schnittstelle für Administratoren zur Ausführung von Aufgaben im Zusammenhang mit Tivoli Inventory über das WWW. Administratoren haben die Möglichkeit, eine Schablone für Benutzerdaten zu erstellen, mit deren Hilfe benutzerspezifische Daten über UserLink for Tivoli Inventory erfasst werden können.

SCS-Komponenten

Alle Umgebungen, die SCS zur Datenerfassung in Tivoli Inventory verwenden, müssen über folgende Komponenten verfügen:

- *Repeater-Stationen*, die in einer *Repeater-Hierarchie* organisiert sind, d. h. Systeme, die den *Multiplexverteilungs*-Service (MDist 2-Service) verwenden. MDist 2-Parameter steuern die Datenverteilung innerhalb einer Tivoli-Umgebung. Über die Repeater-Hierarchie wird die Reihenfolge festgelegt, in der Informationen von einem Repeater zum anderen und schließlich zu den Endpunkten, die Zieladressen der verteilten Daten sind, fließen.

  SCS verwendet eine *Kollektorhierarchie*, die die MDist 2-Repeater-Hierarchie spiegelt. SCS führt aufwärts gerichtete Datenverteilungen innerhalb der Hierarchie aus, während MDist 2 die Daten in der entgegengesetzten Richtung verteilt.

  Weitere Informationen zum Konfigurieren einer Repeater-Hierarchie können Sie dem Handbuch *Tivoli Management Framework: Einsatzplanung* entnehmen.

- *Kollektoren* sind Repeater-Stationen, auf denen SCS installiert ist. Bei einem Kollektor handelt es sich um einen SCS-Dämonprozess auf einem verwalteten Knoten oder einem Gateway, der Daten speichert und dann an andere Kollektoren bzw. an den Inventory Data Handler übermittelt. In diesem Handbuch werden alle verwalteten Knoten als Kollektoren bezeichnet, wenn SCS auf diesen Knoten installiert ist und die Knoten der SCS-Erfassungshierarchie hinzugefügt wurden.

  Kollektoren setzen sich aus folgenden Komponenten zusammen:

  - Das *Depot*, das permanent Daten speichert, die von Endpunkten bzw. anderen Kollektoren abgerufen wurden. Das Depot sendet Daten an andere Kollektoren, wenn sie von diesen angefordert werden.

  - Die *Warteschlangen*, in denen sich die *CTOCs* (Collection Table of Contents) befinden. CTOCs enthalten unter anderem den Namen und die Größe der Datendatei sowie die Quelle und die Zieladresse der erfassten Daten. Die Eingabewarteschlange steuert die Reihenfolge, in der CTOCs zur Datenerfassung verarbeitet werden. Die Ausgabewarte-

schlange steuert die Reihenfolge, in der der Kollektor die CTOCs sendet. Die Warteschlangen für beendete, verzögerte und fehlerhafte Datenerfassungsvorgänge enthalten CTOCs für beendete und verzögerte bzw. Fehlerbedingungen.

- Ein multithreading-fähiger Scheduler-Dämon, der Eingabe- und Ausgabewarteschlangen verarbeitet und den Datenfluss über das Kollektordepot steuert.

- Ein *Kollektor-Manager*, der die Kollektorhierarchie verwaltet, die auf den Daten der Repeater-Hierarchie basiert, die vom MDist 2-Repeater-Manager empfangen wurden.
- Der *Inventory Data Handler* ist das Tivoli Inventory-Objekt, das Daten von Kollektoren empfängt und die Daten an ein oder mehrere RIM-Objekte sendet. (In Tivoli Inventory Version 3.6.2 hieß dieses Objekt Inventory Receiver.) Der Inventory Data Handler ist der letzte Kollektor in einem Tivoli Inventory-System. Er verfügt wie Kollektoren über ein Depot und Warteschlangen. Der Inventory Data Handler dekomprimiert und decodiert die Daten und sendet sie an ein oder mehrere RIM-Objekte anstatt eine Anforderung zur Datenerfassung an einen Kollektor der höheren Ebene abzusetzen.
- Der *Statuskollektor*, der während eines Suchlaufs Statusinformationen für alle Endpunkte erfasst, speichert und verteilt. Sie können den Statuskollektor so konfigurieren, dass dieser Listen mit beendeten Suchläufen, erfolgreichen Suchläufen, fehlgeschlagenen Suchläufen und Fehlernachrichten verwaltet. Der Statuskollektor verwaltet diese Informationen während des gesamten Suchlaufs, so dass Statusinformationen zum Suchlauf während des Suchlaufs verfügbar sind.

  Der Statuskollektor wird auf demselben verwalteten Knoten wie der Inventory Data Handler installiert. Wenn dieser Knoten fehlschlägt und erneut gestartet wird, werden die vom Statuskollektor protokollierten Statusinformationen automatisch wiederhergestellt.

- Das *Callback-Objekt von Inventory*, das die folgenden Funktionen ausführt:
  - Wenn Inventory-Daten über die Kollektorhierarchie nicht erfasst werden können, sendet MDist 2 die Daten an das Callback-Objekt von Inventory, das die Daten an den Inventory Data Handler sendet. Die Daten werden dann über ein oder mehrere RIM-Objekte an das Konfigurations-Repository gesendet.
  - Bei allen Suchläufen sendet MDist 2 eine Statusnachricht an das Callback-Objekt von Inventory, die anzeigt, ob das Inventory-Profil erfolgreich zum Endpunkt gesendet werden konnte. Diese Nachricht zeigt nur an, dass der Endpunkt das Profil verarbeitet hat. Ob die Suchlaufdaten das Konfigurations-Repository erreicht haben, wird nicht angezeigt.
- Ein oder mehrere RIM-Objekte, die für den Zugriff auf das Konfigurations-Repository von Tivoli Inventory eine Verbindung zwischen Tivoli Inventory und RDBMS herstellen. Sie können mehrere RIM-Objekte konfigurieren, so dass diese gleichzeitig SCS-Daten in das Konfigurations-Repository schreiben.

Funktionsweise von Tivoli Inventory

Tivoli Inventory setzt auf den von Tivoli Management Framework zur Verfügung gestellten Ressourcen und Verwaltungsstrukturen auf. Von Tivoli Inventory wird das Prinzip der *Verwaltung durch Subskription* unterstützt, bei dem Netzressourcen über die Erstellung verschiedener Profile und deren Verteilung (über den *Profilmanager*) an physische Entitäten (die so genannten Subskribenten) verwaltet werden. Profile sind Container für anwendungsspezifische (in diesem Fall Tivoli Inventory-spezifische) Informationen, während Profilmanager verschiedene Profile enthalten, die an dieselben Subskribenten bzw. Systeme verteilt werden.

Gültige Ziele für die Verteilung eines Inventory-Profils sind Profilmanager und *Endpunkte*. Hierbei handelt es sich um ein System, das über ein anderes System, einem so genannten Gateway, verwaltet wird. Es wird als Ziel für bestimmte Operationen, beispielsweise den Empfang von Profilen oder die Ausführung von Aufgaben, angegeben.

Für den Einsatz von Tivoli Inventory müssen Sie zunächst mindestens ein Inventory-Profil in einem Profilmanager erstellen und anpassen. Über das Fenster des Profilmanagers wird das Inventory-Profil an Subskribenten verteilt. Anschließend führt das Profil auf jedem Zielsystem eine Aktion aus. Das Profil kann eine Konfigurationsdatei an das System verteilen oder einen Suchlauf auf dem System ausführen bzw. je nach Profileinstellungen beide Aktionen ausführen.

Die Suchsoftware erstellt *MIF-Dateien (MIF = Management Information Format)*. Dabei handelt es sich um ASCII-Dateien mit Informationen in einem Standardformat, die bei Abfragen von Hardware- oder Softwareinformationen bzw. benutzerspezifischen Daten erfasst werden. Anschließend verarbeitet Tivoli Inventory die Informationen in den MIF-Dateien und sendet die Daten unter Verwendung von SCS bzw. MDist 2 an das Konfigurations-Repository von Tivoli Inventory.

Diese Daten können Sie über die Abfragefunktion von Tivoli Management Framework anzeigen, sobald das Konfigurations-Repository von Inventory aktualisiert worden ist. So können Sie z. B. anhand des Konfigurations-Repositorys feststellen, welche Systeme über eine alte Version eines Softwareprodukts verfügen, die aufgerüstet werden muss.

Tivoli Inventory steuert zusammen mit SCS, zu welchem Zeitpunkt wie viele Daten in Ihrem Netzwerk erfasst werden.

Tivoli Inventory mit SCS (Scalable Collection Service) verwenden

Dieser Abschnitt enthält eine Beschreibung von SCS, eine Auflistung der Komponenten einer SCS-Umgebung, eine Beschreibung des Zusammenspiels zwischen SCS und Tivoli Inventory bei der Datenerfassung und eine Auflistung der Funktionen und Vorteile von SCS.

Skalierbare Erfassung

Mit Hilfe der skalierbaren Erfassung können Sie steuern, wie viele Daten zu welchem Zeitpunkt in Ihrem Netzwerk erfasst werden. Durch die skalierbare Datenerfassung können Sie den Zeitaufwand für das Durchsuchen und Speichern großer Datenmengen deutlich reduzieren. Mit Hilfe von SCS können Sie die Datenerfassung skalieren.

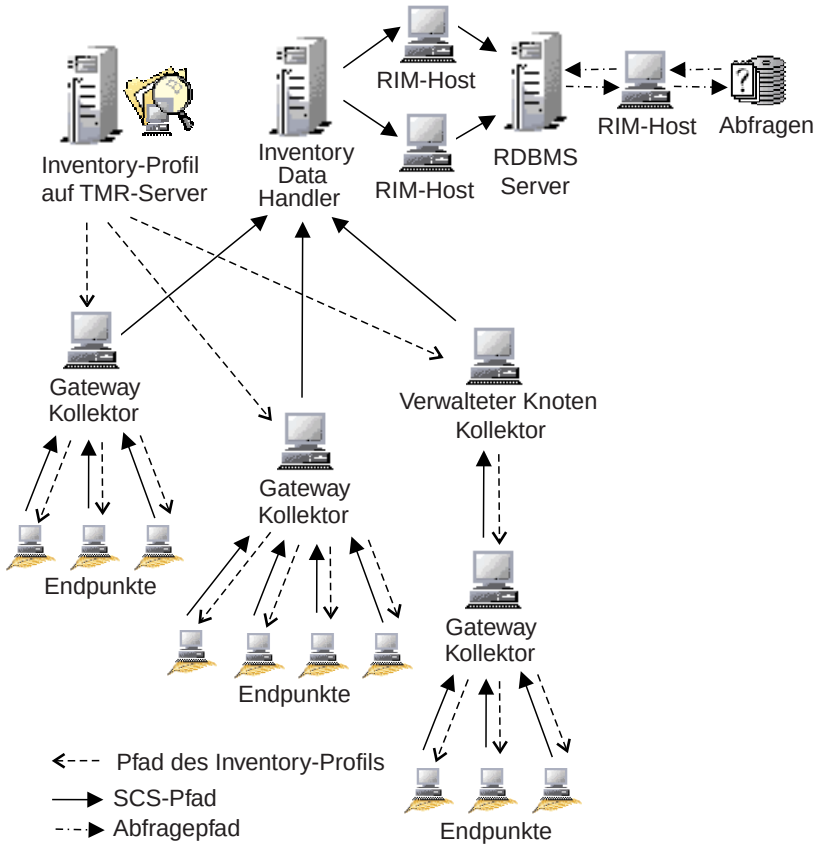
Bei Beendigung des Suchlaufs auf den jeweiligen Endpunkten sendet SCS Suchlaufdaten an die Zieladressen. Innerhalb des Netzwerks werden also kleinere Datenblöcke versendet. Ein weiterer Vorteil von SCS besteht in der Möglichkeit, Daten auf hierarchisch angeordneten Kollektorknoten zwischenzuspeichern. Dadurch verteilt sich die Systembelastung innerhalb der TMR. Mit Hilfe von SCS wird der Datenstrom effektiver gesteuert, da mehrere Pfade zum Konfigurations-Repository erstellt werden. Dank dieser Funktionen können Suchläufe schneller ausgeführt und Netzwerkressourcen besser gesteuert werden.

SCS wird von einem Endpunkt ausgelöst, der über zu erfassende Daten verfügt. Dieser Endpunkt teilt dem Gateway mit, dass die Daten abrufbereit sind, indem er das CTOC sendet. Ein Kollektordämon, der auf dem Gateway ausgeführt wird, ruft anschließend die Daten vom Endpunkt ab und speichert sie auf dem Kollektor in einem permanenten Speicher. Je nach Konfiguration Ihres Netzwerks werden die Daten vor Erreichen ihrer Zieladresse innerhalb einer Kollektorhierarchie weitergegeben.

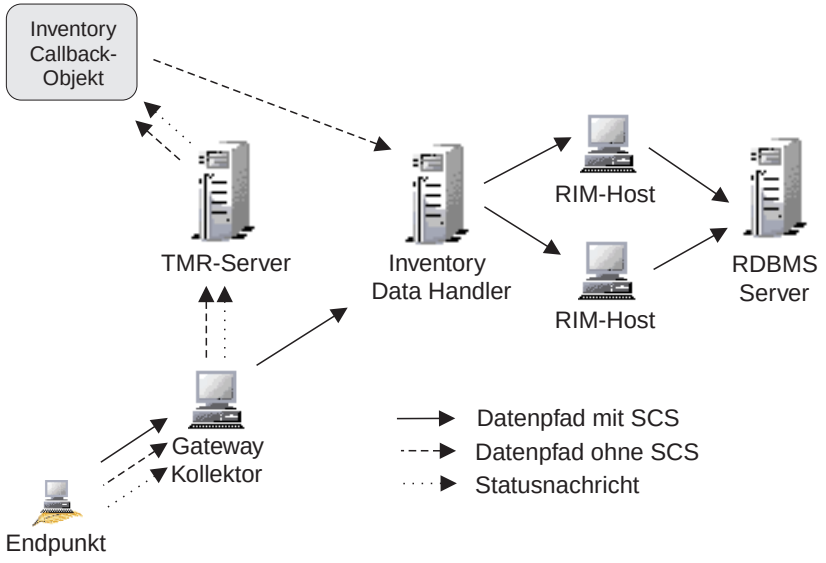
In Tivoli Inventory senden alle Kollektoren Daten an den Inventory Data Handler, der die Daten an das Konfigurations-Repository sendet.

Datenerfassung mit SCS

In folgendem Schaubild wird das Zusammenspiel zwischen SCS und Tivoli Inventory bei der Ausführung von Suchläufen und dem Speichern von Informationen im Konfigurations-Repository dargestellt.



In folgendem Schaubild wird die Funktion des Callback-Objekts von Inventory verdeutlicht.



Die Datenerfassung mit Tivoli Inventory und SCS ist in drei Hauptphasen untergliedert:

- Zunächst durchsucht das Inventory-Profil Endpunkte. Die Endpunkte senden daraufhin CTOCs an SCS.
- Anschließend verarbeitet SCS die CTOCs und verschiebt die Daten innerhalb der Kollektorhierarchie in den Inventory Data Handler.
- Schließlich schreibt der Inventory Data Handler die Daten in ein oder mehrere RIM-Objekte, die die Daten wiederum an das Konfigurations-Repository senden.

Diese Phasen werden in den folgenden Absätzen beschrieben.

In der ersten Phase verteilen Sie das Profil an die zu durchsuchenden Endpunkte. Sobald der Suchlauf auf allen Endpunkten abgeschlossen ist, wird von Tivoli Inventory eine komprimierte und codierte Datendatei generiert.

Der Endpunkt sendet ein CTOC, das Informationen zur Datendatei enthält, an den Kollektor auf dem Gateway, der den Endpunkt verwaltet. Der Kollektordämon reiht das CTOC zur Verarbeitung in die Warteschlange ein.

In der zweiten Phase verschiebt SCS die Daten von allen Endpunkten in ein Depot im Gateway-Kollektor, der den Endpunkt steuert. Während die Daten vom Endpunkt in das Depot verschoben werden, verbleibt das CTOC für die Daten in der Eingabewarteschlange des Kollektors. Wenn der Kollektor alle Daten vom Endpunkt abgerufen hat, wird das CTOC in die Ausgabewarteschlange gestellt. Daraufhin teilt der Kollektor dem nächsten Kollektor in der Hierarchie mit, dass die Daten abrufbereit sind. Der Kollektor, der sich auf einer höheren Ebene in der Hierarchie befindet, stellt das CTOC in seine Eingabewarteschlange und ruft dann die Daten von dem Kollektor, der sich auf einer niedrigeren Ebene in der Hierarchie befindet, ab. Wenn die Datenübertragung an den Kollektor der höheren Ebene abgeschlossen ist, löscht der Kollektor der niedrigeren Ebene das CTOC in seiner Ausgabewarteschlange und die Daten in seinem Depot.

Dieser Austausch findet zwischen allen Kollektoren in der Hierarchie statt, da die Kollektoren die Daten an den Inventory Data Handler asynchron übermitteln.

In der dritten Phase sendet der Inventory Data Handler die erfassten Daten an ein beliebiges verfügbares RIM-Objekt. Die erfassten Daten werden vom RIM-Objekt an das Konfigurations-Repository von Tivoli Inventory gesendet. Der Inventory Data Handler sendet den Fertigstellungsstatus aller durchsuchten Endpunkte sowie Fehlermeldungen an den Statuskollektor . Der Statuskollektor ändert für alle Endpunkte den Status ’Anstehend’ in ’Erfolgreich’ oder ’Fehlgeschlagen’.

Tivoli Inventory- und SCS-Funktionen

Neben der Erfassung von Informationen zu unternehmensinternen Systemen ermöglicht Tivoli Inventory außerdem eine zentrale Systemverwaltung in verteilten Client-/Server-Umgebungen. In den folgenden Abschnitten werden die Funktionen von Tivoli Inventory und SCS beschrieben.

Tivoli Inventory-Funktionen

Tivoli Inventory bietet folgende Funktionen:

- Eine flexible, standardisierte Architektur für die Datenspeicherung. Tivoli Inventory sichert die während eines Inventory-Suchlaufs erfassten Daten in MIF-Dateien, die dem DMTF-Standard (*DMTF = Desktop Management Task Force*) 2.0 entsprechen. Darüber hinaus wird der Inhalt der MIF-Dateien im Konfigurations-Repository gespeichert.

  Anmerkung: Bei DMTF handelt es sich um ein Konsortium, das sich um die Definition einer Standardsyntax für Daten bemüht, über die die Verwaltung der Komponenten auf PCs festgelegt wird.

- Umfangreiche Optionen zur Anpassung von Inventory-Profilen. Sie haben die Möglichkeit, Inventory-Profile so zu konfigurieren, dass sie bei ihrer Verteilung an die Zielsysteme folgende Vorgänge ausführen:

  - Hardware- und Softwareabfragen - Die Zielsysteme können von Tivoli Inventory nach Hardware- und/oder Softwareinformationen durchsucht werden. Softwaresuchläufe können dahingehend konfiguriert werden, dass bestimmte Verzeichnisse und Dateien durchsucht werden.

    Auf PCs werden bei Hardwaresuchläufen neben Hardwarekomponenten automatisch auch BIOS-Informationen abgefragt. Darüber hinaus können Sie auch nach DMI-Informationen (DMI = Desktop Management Information) suchen.

  - Starten von Skripts - Sie können Prozeduren in Inventory-Profile aufnehmen, die während der Profilverteilung auf den Zielsystemen ausgeführt werden. Neben vielem anderen kön-

nen Prozeduren benutzerdefinierte Suchsoftware ausführen, mit der Informationen erfasst und in benutzerspezifischen MIF-Dateien gespeichert werden.

- Lesen von Hardware- und Softwaredateien sowie benutzerspezifischen MIF-Dateien - Mit Tivoli Inventory können sowohl MIF-Dateien, die auf Grund eines Hardware- bzw. Softwaresuchlaufs erstellt wurden, als auch vom Benutzer erstellte MIF-Dateien gelesen werden.

Weitere Informationen zum Erstellen, Anpassen und Verteilen von Inventory-Profilen können Sie den Kapiteln „Mit Inventory-Profilen arbeiten” auf Seite 163 und „Inventory-Profile verteilen” auf Seite 197 entnehmen.

- Konfigurierbare Hardwaresuchläufe. Sie können den Datentyp, der von einem Hardwaresuchlauf zurückgegeben wird, auswählen und die Daten, die von dem Suchlauf generiert werden, reduzieren. Durch Reduzierung der zurückgegebenen Datenmenge können Sie die Geschwindigkeit erhöhen, mit der der Suchlauf ausgeführt wird, die für den Datentransport erforderliche Netzbandbreite verringern und den Zeitaufwand für das Schreiben der Daten in das Konfigurations-Repository herabsetzen.
- WMI-Suchprogramm (Windows Management Instrumentation). In Tivoli Inventory können Daten von Endpunkten unter Verwendung von WMI abgerufen werden, um die Daten zu empfangen, die im CIM (Common Information Model) auf Windows-Plattformen, die CIM und WMI für allgemeine Inventory-Attribute unterstützen, gespeichert sind. Diese Funktion wird automatisch verwendet, wenn sie verfügbar ist.
- Abfragen, die von Endpunkten eingeleitet werden. Sie können eine Abfrage von einem Endpunkt anstatt von einem Inventory-Profil einleiten. Diese Funktion ist in vielerlei Hinsicht nützlich. Sie können beispielsweise Abfragen von Endpunkten aus einleiten, um folgende Aktionen auszuführen:
  - Maschine bei jedem Neustart durchsuchen
  - Laptop bei jeder Anmeldung eines mobilen Benutzers am Netzwerk durchsuchen

- Maschine, die vom Netzwerk getrennt ist, durchsuchen

- Ein Inventory-Abfragedialog, mit dessen Hilfe Sie bei Angabe von Auswahlkriterien das Konfigurations-Repository nach einem bestimmten System durchsuchen können. Weitere Informationen hierzu können Sie dem Abschnitt „Inventory-Daten abfragen” auf Seite 209 entnehmen.

- Protokollierung. Sie können Protokolltabellen erstellen, in denen alle Änderungen an den Suchlaufdaten im Konfigurations-Repository von Tivoli Inventory aufgezeichnet werden. Wenn Sie diese Tabellen nicht erstellen, werden Änderungen an den Suchlaufdaten von Tivoli Inventory nicht protokolliert. Weitere Informationen können Sie dem Abschnitt „Protokollfunktion” auf Seite 366 entnehmen.

- Eine Web-basierte Schnittstelle. Mit Hilfe eines Web-Browsers oder über die Befehlszeile können Sie die Benutzerdatenschablone für die Erstellung eines Benutzerdatenformulars verwenden. Auf der Basis dieser Benutzerdatenformulare ist es möglich, MIF-Dateien mit den benutzerspezifischen Informationen, die bei späteren Suchläufen erfasst werden, zu erstellen. Beispielsweise können Benutzer Raumnummern und Telefonnummern eingeben.

  Weitere Informationen zu Web-Seiten von Tivoli Inventory können Sie dem Kapitel „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121, entnehmen.

- Unterstützung von mobilen Endpunkten. Sie können ein Profil für einen benutzerspezifischen Inventory-Suchlauf erstellen, das an einen mobilen bzw. fernen Endpunkt im Push-Modus gesendet werden kann. Sie können festlegen, dass der Suchlauf ausgeführt wird, wenn die Verbindung zum Endpunkt unterbrochen ist; die Daten werden gesendet sobald die Verbindung des Endpunkts zum Netzwerk wiederhergestellt ist.

- Wake-on-LAN (WoL)-Technologie. Mit Hilfe der MDist 2-Funktionen können Systeme, die über eine WoL-fähige Netzwerkkarte nach der Wired for Management-Spezifikation von Intel verfügen, automatisch von Tivoli Inventory aktiviert werden.

- Einsatz zusammen mit anderen Tivoli-Anwendungen. Sie haben die Möglichkeit, Tivoli Inventory zusammen mit der Abfragefunktion von Tivoli Management Framework sowie der Komponente Tivoli Software Distribution für Systemverwaltungs-Tasks einzusetzen. Weitere Informationen finden Sie unter „Tivoli Inventory und andere Tivoli-Anwendungen” auf Seite 16.

Funktionen von SCS

Die folgenden Leistungsmerkmale von SCS zeigen die Vorteile der asynchronen Datenerfassung:

- Durch die asynchrone Datenerfassung und die asynchrone Verarbeitung können die Prozesse besser auf mehrere Knoten in der TMR verteilt werden. Dadurch wird das Netzwerk effektiver genutzt. Die Daten werden auf ihrem Weg zum Konfigurations-Repository von SCS auf verschiedenen Kollektoren zwischengespeichert. Dadurch kann der Datenfluss zwischen aufeinanderfolgenden Kollektoren in der Hierarchie gesteuert werden.
- Die Suchlaufdaten werden von SCS zurückgegeben, sobald der Suchlauf auf allen Endpunkten abgeschlossen ist. Auf Grund dieser Funktion wird das RIM-Objekt am Ende einer Verteilung nicht überbelastet.
- Wenn die Daten von einem Endpunkt abgerufen wurden, kann die Verbindung zu diesem Endpunkt unterbrochen werden, ohne dass sich dies auf SCS auswirkt.
- Der Inventory Data Handler kann Daten gleichzeitig an ein oder mehrere RIM-Objekte schreiben, das bzw. die über eine oder mehrere Verbindungen verfügen.
- SCS verringert die Belastung für den TMR-Server. Die Suchlaufdaten für einzelne Endpunkte werden vom Inventory Data Handler zwischengespeichert, während das Konfigurations-Repository aktualisiert wird. (Es wird empfohlen, dass Sie den Inventory Data Handler auf einem anderen System als dem TMR-Server erstellen).
- Mit Hilfe von SCS und dem Tivoli-Scheduler können Sie Suchläufe und den Datenaustausch im Netzwerk für Zeiträume terminieren, in denen wenig Daten im Netzwerk ausgetauscht werden.

- Kollektoren innerhalb der Kollektorhierarchie speichern die erfassten Daten in Depots. Beim Auftreten eines Fehlers kann der Kollektor die Daten erneut senden, wenn er wieder online ist. Die Endpunkte müssen also nicht erneut durchsucht werden.
- SCS bietet Funktionen, über die Sie Informationen zum Status der Kollektoren, zu CTOCs und Suchläufen abfragen können.
- Mit Hilfe von SCS können Sie den Status einzelner Ziele feststellen, sobald der Vorgang für diese abgeschlossen wurde. Sie müssen nicht warten bis der Vorgang auf allen Zielen abgeschlossen ist.
- Nach Auftreten eines Fehlers können Sie mit SCS erneut versuchen, Daten von Endpunkten bzw. anderen Kollektoren abzurufen; Sie können die Anzahl der Wiederholungen festlegen.

Weitere Informationen zu den Funktionen von SCS können Sie dem Kapitel „Mit SCS arbeiten” auf Seite 91, entnehmen.

Tivoli Inventory und andere Tivoli-Anwendungen

Tivoli Inventory ist ein weiterer Schritt in Richtung einer *umfassenden Verwaltungsumgebung*, ein Ansatz, der bei der Tivoli-Verwaltungsumgebung verfolgt wird. In einem einzigen verwaltungstechnischen Vorgang können Sie nicht berechtigte Änderungen an Softwarekonfigurationen aufspüren und die ursprünglichen Einstellungen mit Hilfe von Tivoli Inventory-, Tivoli Software Distribution- und Tivoli Management Framework-Tasks wiederherstellen. Der gemeinsame Einsatz von Anwendungen und Tools ermöglicht unter anderem die Ausführung folgender Operationen:

- Bei einem gemeinsamen Einsatz von Tivoli Inventory und Tivoli Management Framework erhalten Administratoren automatisch eine entsprechende Nachricht, falls auf einem unternehmensinternen System eine Anwendung hinzugefügt wird, für die keine Berechtigung vorliegt.

- Bei einem gemeinsamen Einsatz von Tivoli Inventory, Tivoli Software Distribution und Tivoli Management Framework können die Softwarekonfigurationen der einzelnen Systeme daraufhin überprüft werden, ob wichtige Dateien fehlen; ist dies der Fall, kann die ursprüngliche Konfiguration wiederhergestellt werden. Nach der Erstellung und Installation verwaltbarer Anwendungen ist ein Abgleich aller Anwendungen und Systemkonfigurationen im gesamten Unternehmen möglich.
- Bei einem gemeinsamen Einsatz von Tivoli Inventory und der Abfragefunktion von Tivoli Management Framework können Sie zur Leistungsverrechnung die gesamte Systemhardware- und -software kontrollieren und abfragen.

Sicherheit

Für den Einsatz von Tivoli Inventory gelten die Sicherheitsbestimmungen von Tivoli Management Framework, um so die sichere Verwendung von Anwendungen zu gewährleisten und unbefugten Zugriffen vorzubeugen. Dies bedeutet, dass allen Administratoren, die mit Tivoli Inventory arbeiten möchten, eine entsprechende Tivoli-Sicherheitsberechtigung (auch Berechtigungsklasse genannt) erteilt werden muss; andernfalls ist die Ausführung von Inventory-Tasks nicht möglich. Weitere Informationen zum Zuordnen von Berechtigungsklassen können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

Vielen in diesem Handbuch vorgestellten Prozeduren ist eine Tabelle mit der jeweils erforderlichen Berechtigungsklasse vorangestellt. Weitere Informationen zu den für die einzelnen Operationen erforderlichen Berechtigungsklassen können Sie dem Abschnitt „Berechtigungsklassen” auf Seite 235 entnehmen.

2

Tivoli Inventory Version 4.0 installieren

In diesem Kapitel werden die einzelnen Schritte erläutert, die bei der Installation von Tivoli Inventory Version 4.0 in Ihrer Tivoli-Umgebung erforderlich sind.

Übersicht

Führen Sie vor der Installation von Tivoli Inventory folgende Schritte aus:

1. Lesen Sie die *Release-Notes für Tivoli Inventory Version 4.0*.
2. Installieren und konfigurieren Sie Tivoli Management Framework Version 3.7.1 auf dem TMR-Server. Stellen Sie sicher, dass alle Systeme, für die Sie einen Suchlauf durchführen möchten, Endpunkte in der TMR sind. Weitere Informationen können Sie dem Handbuch *Tivoli Management Framework: Einsatzplanung*, dem *Tivoli Management Framework Benutzerhandbuch* und den *Tivoli Management Framework Release-Notes* entnehmen.
3. Stellen Sie sicher, dass alle Systeme, auf denen die Funktionen von UserLink for Tivoli Inventory angewendet werden sollen, über einen Web-Browser verfügen. Anweisungen zur Installation können Sie den Handbüchern zu dem entsprechenden Web-Browser entnehmen. Eine Liste der unterstützten Web-Browser finden Sie in den *Tivoli Inventory-Release-Hinweisen*.

4. Sichern Sie die Tivoli Management Framework-Objektdatenbank für alle Systeme innerhalb der TMR.

Im Folgenden werden kurz die einzelnen Schritte aufgeführt, die für die Installation und Konfiguration von Tivoli Inventory Version 4.0 erforderlich sind.

1. Installieren Sie SCS (Scalable Collection Service). Dieser Service wird als Programmkorrektur für Tivoli Management Framework installiert. Weitere Informationen finden Sie im Abschnitt „SCS (Scalable Collection Service) installieren" auf Seite 22.
2. Installieren und konfigurieren Sie ein Verwaltungssystem für relationale Datenbanken für die Kommunikation mit dem RIM-Host. *Das Verwaltungssystem für relationale Datenbanken muss vor der Installation von Inventory installiert werden*. Siehe hierzu „Verwaltungssystem für relationale Datenbanken (RDBMS) konfigurieren" auf Seite 31. Eine Liste der unterstützten Verwaltungssysteme für relationale Datenbanken können Sie den *Tivoli Management Framework-Release-Hinweisen Version 3.7.1* entnehmen.
3. Installieren Sie die Tivoli Inventory Server-Anwendung auf dem TMR-Server und auf den verwalteten Knoten. Siehe hierzu „Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren" auf Seite 43.
4. Installieren Sie die Tivoli Inventory Gateway-Anwendung auf den verwalteten Knoten, die unter Tivoli Management Framework als Gateways konfiguriert wurden. Siehe hierzu „Tivoli Inventory Gateway installieren" auf Seite 55.

Dieses Kapitel enthält außerdem Hinweise zur Deinstallation von Tivoli Inventory Version 4.0. Siehe hierzu „Tivoli Inventory deinstallieren" auf Seite 56.

Nach der Installation der Tivoli Inventory-Produkte auf den Maschinen in Ihren TMRs sind folgende Schritte erforderlich, um die von Tivoli Inventory erfassten Daten speichern und abrufen zu können. Eine Erläuterung der folgenden Aufgaben können Sie dem Kapitel „Konfigurations-Repository erstellen" auf Seite 63 entnehmen.

1. Erstellen Sie das Konfigurations-Repositorys; siehe hierzu „Konfigurations-Repository erstellen” auf Seite 63.
2. Installieren Sie die Liste der standardmäßigen Softwaresignaturen im Konfigurations-Repository; siehe hierzu „Softwaresignaturen installieren” auf Seite 85.
3. Erstellen Sie die Tivoli Inventory-Abfragen; siehe hierzu „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87.

Koexistenz mit früheren Versionen des Produkts

Tivoli Inventory Version 4.0 ist ein neues Produkt und kein Produkt zur Aufrüstung von Tivoli Inventory Version 3.6.*x*. Für dieses neue Produkt ist die Installation früherer Versionen von Tivoli Inventory nicht erforderlich. Die Koexistenz von Tivoli Inventory 4.0 und früheren Versionen des Produkt ist jedoch möglich. Aus folgenden Gründen ist die Installation der neuen Version zusammen mit früheren Versionen von Tivoli Inventory sinnvoll:

- Um die Migration von der vorherigen Version zur neuen Version einfacher zu gestalten.
- Damit Sie in Ihrer derzeitigen, angepassten Umgebung arbeiten und gleichzeitig die neuen Funktionen, die Tivoli Inventory Version 4.0 bietet, nutzen können.

Weitere Informationen zur Konfiguration von Tivoli Inventory Version 4.0 zur gemeinsamen Einsatz mit früheren Versionen können Sie den *Tivoli Inventory-Release-Hinweisen* entnehmen.

Tivoli Software Installation Service verwenden

Bei dem Programm Tivoli Software Installation Service (SIS) handelt es sich um ein Produkt, mit dem mehrere Tivoli-Programme auf mehreren Systemen gleichzeitig installiert werden können. Dieses Programm basiert auf Java, daher ist eine Installation mehrerer Programme auf mehreren Systemen wesentlich schneller, als dies mit dem Installationsprogramm von Tivoli Management Framework der Fall ist. SIS führt Überprüfungen der Produktvoraussetzungen sowie, falls angegeben, Überprüfungen von benutzerdefinierten Vorausset-

zungen aus und minimiert dadurch mögliche Installationsfehler. Dadurch treten Fehler nur noch dann auf, wenn Systeme ausgeschaltet oder aus dem Netzwerk entfernt wurden.

SIS erstellt auch ein Installations-Repository (IR), in das das Installationsimage von einem oder mehreren Tivoli-Produkten importiert werden kann. Es können nur die Interpreter importiert werden, die in Ihrer Umgebung benötigt werden; dadurch wird weniger Plattenspeicherplatz belegt und der Importvorgang beschleunigt. Das Installations-Repository stellt die Quelle für alle Tivoli-Installationen dar. Sie haben außerdem die Möglichkeit, ein Installations-Repository in mehreren TMRs gemeinsam zu benutzen.

Es wird empfohlen, eine Aufrüstung des Installationsprogramms in Ihrer aktuellen Tivoli Management Framework-Installation vorzunehmen, indem Sie SIS installieren. Wenn Sie Tivoli Management Framework zum ersten Mal installieren, müssen Sie SIS auf dem ersten verwalteten Knoten installieren, auf dem ein von SIS unterstütztes Betriebssystem läuft. Anschließend können Sie mit Hilfe von SIS weitere Tivoli-Produkte installieren. Anweisungen zur Installation von SIS in Ihrer Tivoli Management Framework-Installation sowie zur Installation weiterer Produkte mit SIS können Sie dem *Tivoli Software Installation Service Benutzerhandbuch* entnehmen.

SCS (Scalable Collection Service) installieren

Durch SCS werden die Funktionen der verwalteten Knoten und der zugehörigen Gateways in Tivoli erweitert. SCS wird als Programmkorrektur für das Tivoli Management Framework-Produkt installiert. Es wird empfohlen, SCS auf allen unterstützten Knoten in allen TMRs zu installieren.

SCS verwendet eine Kollektorhierarchie, um Daten zu erfassen, zwischenzuspeichern und an die Zieladressen zu senden. Die Kollektorhierarchie basiert auf der Repeater-Hierarchie, die von dem Multiplexverteilungs-Service (MDist 2) eingerichtet wurde.

Möglicherweise installieren Sie SCS unter einer der beiden folgenden Voraussetzungen:

- Sie installieren Tivoli-Produkte zum ersten Mal und haben keine Repeater-Hierarchie erstellt.
- Ihre Repeater-Hierarchie ist bereits vorhanden.

Wenn Sie keine Repeater-Hierarchie erstellt haben, verwendet SCS eine Standardkonfiguration. Gemäß dieser Konfiguration sind alle Gateways Kollektoren, die Daten direkt an den Inventory Data Handler senden.

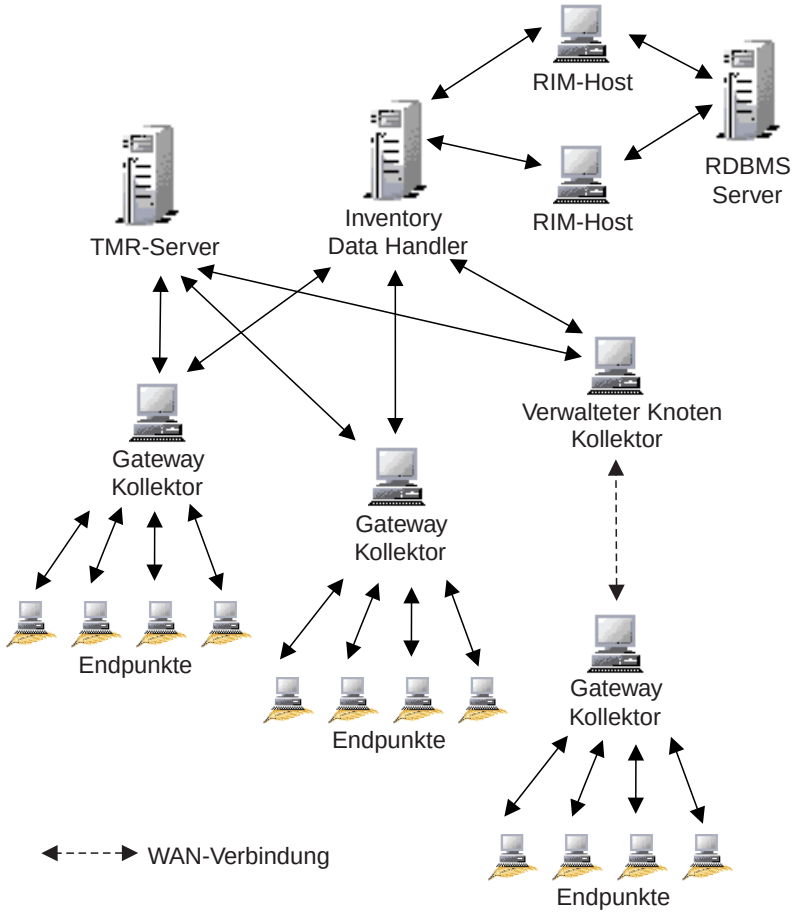
Wenn Sie bereits eine Repeater-Hierarchie erstellt haben, wird die Repeater-Hierarchie von SCS gespiegelt. Wenn Sie Ihrer Hierarchie einen neuen Repeater hinzufügen, müssen Sie SCS auf dem verwalteten Knoten installieren, auf dem sich der neue Repeater befindet, falls Sie dies nicht bereits getan haben.

Da SCS und der MDist 2-Service eine gemeinsame Netzwerk-Topologie haben, sollten Sie die Hierarchie der Kollektoren und Repeater sorgfältig planen. Beispiel: Alle Änderungen, die Sie an der Repeater-Hierarchie vornehmen, werden auf Ihre Datenerfassungs-Hierarchie übertragen. Dasselbe gilt auch im umgekehrten Fall. Aus diesem Grund sollten Sie Ihr Netzwerk für beide Services, SCS und den MDist 2-Service, optimieren.

Anmerkung: Wenn Sie Ihre Repeater-Hierarchie ändern, wird die Kollektorhierarchie erst dann geändert, wenn Sie den Befehl **wcollect** unter Verwendung der Optionen **–r** auf allen betroffenen Kollektoren ausführen oder den TMR-Server und alle Systeme, die als Kollektoren fungieren, mit dem Befehl **odadmin** erneut ausführen. Weitere Informationen zum Befehl **wcollect** können Sie dem Abschnitt „wcollect” auf Seite 247 entnehmen. Weitere Informationen zum Befehl **odadmin** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

In der Regel empfiehlt es sich, ein als Kollektor und Repeater konfiguriertes System an beiden Enden einer kritischen Verbindung in Ihrem Netzwerk wie beispielsweise einer WAN-Verbindung (Wide Area Network) einzusetzen.

Bei dieser Konfiguration müssen die Daten nur von einem System innerhalb des WANs abgerufen werden. In folgendem Schaubild wird diese Konfiguration gezeigt.



Weitere Informationen zum Einrichten einer Repeater-Hierarchie können Sie dem Handbuch *Tivoli Management Framework: Einsatzplanung* entnehmen.

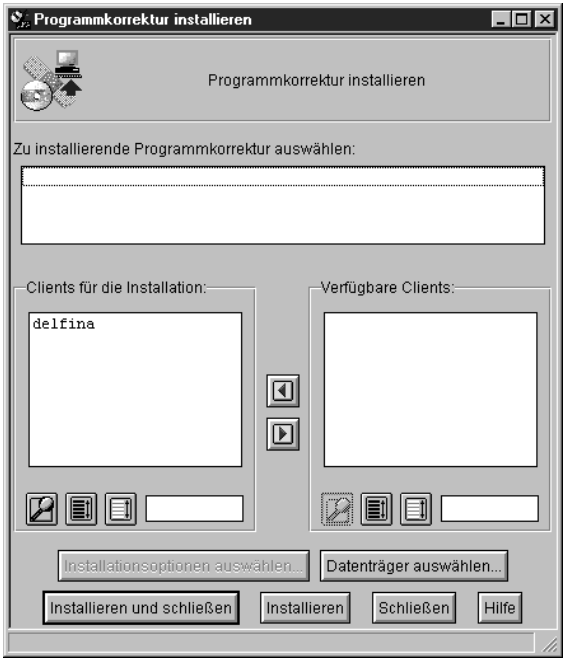
SCS-Programmkorrektur installieren

Die folgenden Abschnitte enthalten Erläuterungen zur Installation von SCS für den gemeinsamen Einsatz mit Tivoli Inventory. Sie müssen SCS auf allen verwalteten Knoten in Ihrer Tivoli-Umgebung installieren.

Datenträger auswählen

Führen Sie folgende Schritte aus, um auf die zu installierenden Dateien über die Tivoli-Arbeitsoberfläche zugreifen zu können:

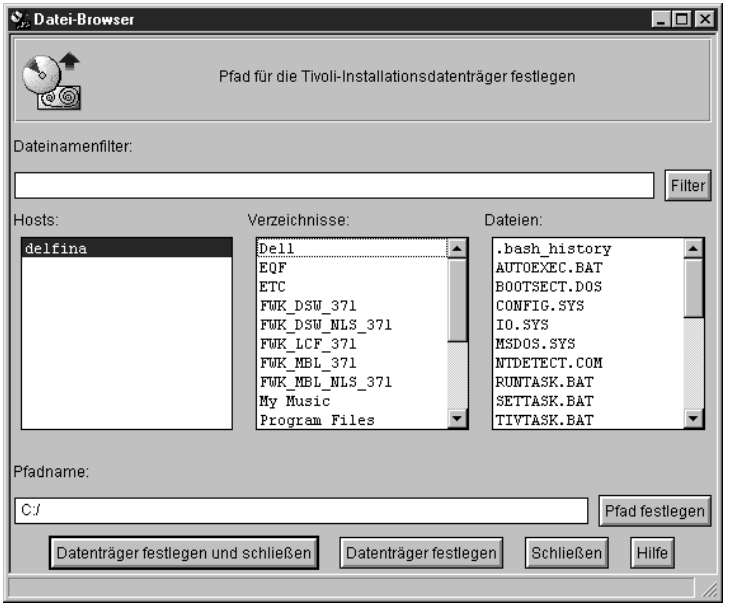
1. Starten Sie die Tivoli-Arbeitsoberfläche auf dem TMR-Server.
2. Wählen Sie im Menü 'Arbeitsoberfläche' nacheinander die Optionen **Installieren** –> **Programmkorrektur installieren** aus. Daraufhin wird der Dialog 'Programmkorrektur installieren' geöffnet.



Falls Tivoli Scalable Collection Service For Framework 3.7.1 in der Liste 'Zu installierende Programmkorrektur auswählen'

bereits enthalten ist, fahren Sie mit Abschnitt „Tivoli Scalable Collection Service For Framework 3.7.1 installieren” auf Seite 27 fort, andernfalls mit Schritt 3.

3. Klicken Sie auf **Datenträger auswählen**. Daraufhin wird der Dialog ’Datei-Browser’ geöffnet.

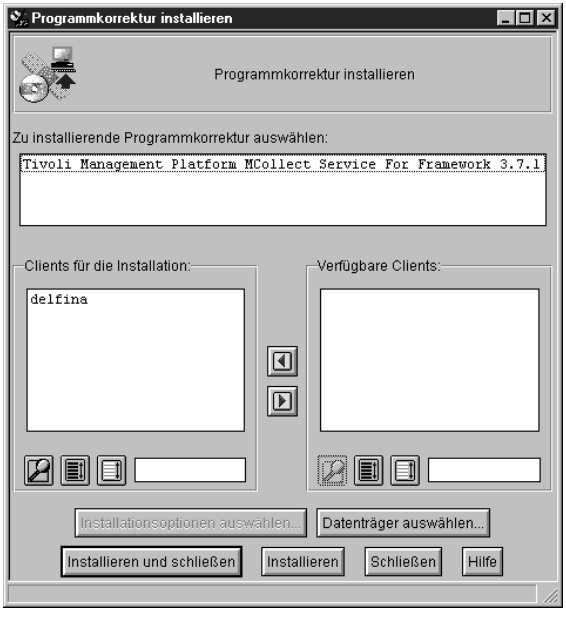


4. Geben Sie im Feld ’Pfadname’ den Pfadnamen der CD-ROM mit der Tivoli-Software mit Hilfe einer der folgenden Methoden ein:
   - Geben Sie im Feld ’Pfadname’ den vollständigen Pfad ein.
   - Durchsuchen Sie das Dateisystem wie folgt:
     a. Wählen Sie in der Liste ’Hosts’ den Host (bzw. das Laufwerk) aus, auf dem sich die CD-ROM befindet. Nach der Auswahl eines Hosts wird die Liste ’Verzeichnisse’ aktualisiert und zeigt die Verzeichnisse (unter dem Stammverzeichnis) auf dem von Ihnen ausgewählten Host an.

b. Doppelklicken Sie in der Liste ’Verzeichnisse’ auf das Verzeichnis, in dem sich der Installationsdatenträger befindet. Bei Auswahl eines Verzeichnisses wird die Liste ’Dateien’ aktualisiert.

c. Klicken Sie auf **Pfad festlegen**.

5. Klicken Sie auf **Datenträger festlegen und schließen**; daraufhin kehren Sie in den Dialog ’Programmkorrektur installieren’ zurück . Tivoli Scalable Collection Service For Framework 3.7.1 wird in der Liste ’Zu installierende Programmkorrektur auswählen’ angezeigt.



Tivoli Scalable Collection Service For Framework 3.7.1 installieren

Führen Sie die folgenden Schritte aus, um die SCS-Programmkorrektur Tivoli Scalable Collection Service For Framework 3.7.1 auf verwalteten Knoten, die als Gateways für verwaltende Endpunkte fungieren werden, auf verwalteten Knoten, die als Kollektoren fungieren werden, und auf dem TMR-Server zu installieren:

1. Wählen Sie aus dem Menü ’Arbeitsoberfläche’ nacheinander die Optionen **Installieren** –> **Programmkorrektur installieren** aus. Daraufhin wird der Dialog ’Programmkorrektur installieren’ geöffnet.

2. Wählen Sie im Dialog ’Programmkorrektur installieren’ in der Liste ’Zu installierende Programmkorrektur auswählen’ den Eintrag **Tivoli Scalable Collection Service For Framework 3.7.1** aus.

3. Stellen Sie sicher, dass der TMR-Server und alle verwalteten Knoten, auf denen Sie SCS installieren möchten, in der Liste ’Clients für die Installation’ enthalten sind.

   Anmerkung: Es wird empfohlen, SCS auf allen unterstützten Knoten zu installieren. SCS kann nicht auf OS/390- bzw. OS/2-Systemen installiert werden.

4. Wählen Sie in der Liste ’Clients für die Installation’ alle verwalteten Knoten aus, auf denen Sie SCS *nicht* installieren möchten. Klicken Sie auf den Rechtspfeil, um die ausgewählten verwalteten Knoten in die Liste ’Verfügbare Clients’ zu verschieben.

5. Starten Sie die Installation von SCS, indem Sie auf **Installieren** klicken. Im Dialog 'Programmkorrektur installieren' werden die Vorgänge aufgelistet, die bei der Installation ausgeführt werden; außerdem enthält er Hinweise auf Fehler, die Sie vor der Installation beheben sollten. Sie können eine der folgenden Optionen auswählen:

   - Überprüfen Sie die Statusinformationen, und klicken Sie auf **Installation fortsetzen**. Sobald die Installation beendet ist, wird im Dialog 'Programmkorrektur installieren' eine entsprechende Nachricht angezeigt.
   - Um SCS zu einem späteren Zeitpunkt zu installieren, klicken Sie auf **Abbrechen**.

Über die Befehlszeile

Sie können den Befehl **wpatch** verwenden, um SCS über die Befehlszeile zu installieren. Im folgenden Beispiel wird SCS auf den verwalteten Knoten x, y und z installiert.

```
wpatch -c $IMAGE_PATH/cdrom -i MCOLLECT.IND x y z
```

Weitere Informationen zum Befehl **wpatch** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

RIM und das Verwaltungssystem für relationale Datenbanken verwenden

Tivoli Inventory führt Abfragen von Zielmaschinen durch und speichert die dabei erfassten Daten im Konfigurations-Repository, bei dem es sich um eine RDBMS-Datenbank handelt. Dieses Konfigurations-Repository enthält Informationen zu den Systemkonfigurationen in Ihrer Tivoli-Umgebung. Für die Kommunikation mit dem RDBMS-Server verwendet Tivoli Inventory das RDBMS Interface Module (RIM).

Tivoli Inventory speichert Daten im Konfigurations-Repository, das zusammen mit Tivoli Inventory installiert wird. Tivoli Software Distribution speichert ebenfalls Daten im Konfigurations-Repository. Wenn Sie Tivoli Software Distribution verwenden, können Sie dem Handbuch *Tivoli Software Distribution Benutzerhandbuch* weitere

Informationen zur Verwendung des Konfigurations-Repositorys durch Tivoli Software Distribution entnehmen.

Für den Einsatz des Verwaltungssystems für relationale Datenbanken in Tivoli Inventory müssen Sie zunächst die RDBMS-Client- und -Server-Software konfigurieren, die der RIM-Host für die Kommunikation mit dem Verwaltungssystem für relationale Datenbanken benötigt.

Vor der Installation von Tivoli Inventory müssen Sie zunächst einen RIM-Host und einen RDBMS-Server auswählen. Es wird empfohlen, den TMR-Server, den RDBMS-Server und den RIM-Host jeweils auf einem eigenen System zu installieren.

RIM-Host auswählen

Bei dem RIM-Host handelt es sich um den verwalteten Knoten, auf dem die RIM-Objekte installiert sind. Sie legen den RIM-Host fest, indem Sie dessen Namen bei der Installation von Tivoli Inventory im Feld 'RIM-Host' des Dialogs 'Installationsoptionen' angeben. Eine Beschreibung des RIM-Objekts, das von Tivoli Inventory während der Installation erstellt wird, finden Sie im Abschnitt „Kennwörter für RIM-Objekte ändern" auf Seite 51.

Bei der Auswahl des RIM-Hosts sollten Sie Folgendes berücksichtigen:

- Es muss sich um einen verwalteten Knoten in der TMR handeln.
- Auf dem verwalteten Knoten muss die Software des RDBMS-Clients installiert sein. Weitere Informationen hierzu können Sie dem Abschnitt „Verwaltungssystem für relationale Datenbanken (RDBMS) konfigurieren" auf Seite 31 entnehmen.
- Er kommuniziert mit dem Verwaltungssystem für relationale Datenbanken.
- Falls es sich bei dem RIM-Host um ein System unter HP-UX oder Windows NT handelt, sollten Sie sicherstellen, dass der Benutzer 'tmersrvd' auf diesem System definiert ist. Weitere Informationen zum Benutzer 'tmersrvd' können Sie den *Tivoli Management Framework-Release-Hinweisen* entnehmen.

RDBMS-Server auswählen

Bei dem RDBMS-Server handelt es sich um das System, auf dem Sie das Verwaltungssystem für relationale Datenbanken und das Schema des Konfigurations-Repositorys von Tivoli Inventory installieren. Bei der Auswahl des RDBMS-Server sollten Sie Folgendes berücksichtigen:

- Er muss über eine TCP/IP-Verbindung zum RIM-Host verfügen.
- Es muss sich nicht um einen verwalteten Knoten in der TMR handeln, er muss sich jedoch in demselben Netz wie die TMR befinden.
- Er muss über ausreichend Plattenspeicherplatz für die Informationen verfügen, die im Konfigurations-Repository gespeichert werden sollen. Informationen zum Platzbedarf können Sie den *Tivoli Inventory-Release-Hinweisen* entnehmen.
- Der RDBMS-Server kann von mehreren TMRs und Anwendungen verwendet werden.

Verwaltungssystem für relationale Datenbanken (RDBMS) konfigurieren

Bei der Installation von Tivoli Inventory werden Sie im Dialog 'Installationsoptionen' zur Angabe von Informationen zur Konfiguration des Verwaltungssystem für relationale Datenbanken aufgefordert. Diese Informationen sind für die Erstellung des RIM-Hosts in der Tivoli Management Framework-Objektdatenbank erforderlich, der die Kommunikation zwischen Tivoli Inventory und und dem Verwaltungssystem für relationale Datenbanken ermöglicht. Daher muss ein Verwaltungssystem für relationale Datenbanken noch vor Tivoli Inventory installiert und konfiguriert werden.

Nachdem Sie ein Verwaltungssystem für relationale Datenbanken konfiguriert und Tivoli Inventory in Ihrer TMR installiert haben, installieren Sie das Schema des Konfigurations-Repositorys von Tivoli Inventory, indem Sie die RDBMS-Skripts ausführen, die in

Tivoli Inventory zur Verfügung gestellt werden. Weitere Informationen hierzu können Sie dem Kapitel „Konfigurations-Repository erstellen” auf Seite 63 entnehmen.

Sollen Abfragen für mehrere miteinander verbundene TMRs durchgeführt und die dabei ermittelten Informationen in einer einzigen RDBMS-Datenbank gespeichert werden, sind bei der Installation von Tivoli Inventory in den Feldern im Dialog ’Installationsoptionen’ für jede TMR dieselben Angaben erforderlich. Mit dem Befehl **wgetrim** können Sie die Werte des RIM-Hosts einer TMR überprüfen.

Wurden unterschiedliche Werte angegeben, können Sie die Einstellungen mit Hilfe der Befehle **wsetrim** und **wsetrimpw** ändern. Sobald die TMRs miteinander verbunden sind, können Sie Suchläufe in allen diesen TMRs durchführen. Weitere Informationen zu RIM-Befehlen können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Die folgenden Abschnitte enthalten Anweisungen zur Konfiguration der unterstützten Verwaltungssysteme für relationale Datenbanken.

DB2 konfigurieren

Installieren Sie das Verwaltungssystem für relationale DB2-Datenbanken, und erstellen Sie ein DB2-Exemplar und eine DB2-Datenbank auf dem RDBMS-Server, bevor Sie Tivoli Inventory installieren. Dabei muss die Datenbank als Exemplareigner erstellt werden. Anweisungen zum Erstellen eines DB2-Exemplars und einer DB2-Datenbank können Sie der DB2-Dokumentation entnehmen. Nach Ausführung der Skripts für die Installation des Tivoli Inventory-Schemas wird die von Ihnen erstellte Datenbank als Konfigurations-Repository von Tivoli Inventory verwendet. Weitere Informationen hierzu können Sie dem Abschnitt „DB2-Datenbank als Konfigurations-Repository erstellen” auf Seite 64 entnehmen.

Wählen Sie einen verwalteten Knoten als RIM-Host aus. Weitere Informationen zur Auswahl eines RIM-Hosts können Sie dem Abschnitt „RIM-Host auswählen” auf Seite 30 entnehmen.

Es wird empfohlen, dass Sie für Ihren RDBMS-Server und den RIM-Host jeweils unterschiedliche Systeme verwenden. Da eine Kommunikation zwischen RDBMS-Server und RIM-Host erforderlich ist, muss der RIM-Host über ein DB2-Client-Exemplar und eine DB2-Client-Datenbank verfügen. Außerdem muss der Benutzername des DB2-Client-Exemplars mit dem Benutzernamen des Server-Exemplareigners identisch sein. Der Benutzername des Client-Exemplars für Tivoli Inventory Version 4.0 sollte jedoch nicht mit dem Benutzernamen (falls vorhanden) von Tivoli Inventory Version 3.6.2 identisch sein. Der standardmäßige Benutzername in Tivoli Inventory Version 4.0 lautet 'invtiv'. Folgende Schritte sind zur Installation eines Client-Exemplars und einer Client-Datenbank auf dem RIM-Host erforderlich:

1. Melden Sie sich am RIM-Host unter dem Benutzernamen an, der als Exemplareigner vorgesehen ist. Der Exemplareigner verfügt über die Zugriffsberechtigung für das Konfigurations-Repository von Tivoli Inventory.

   Anmerkung: Ersetzen Sie nach der Installation von Tivoli Inventory und Erstellung des Konfigurations-Repositorys mit Hilfe des Befehls **wsetrimpw** das Standardkennwort 'tivoli' durch das Kennwort des Exemplareigners.

2. Starten Sie auf dem RIM-Host das DB2-Dienstprogramm für interaktives SQL. Geben Sie den folgenden Befehl ein:

   ```
   db2
   ```

3. Erstellen Sie einen neuen Client-Knoten, von dem aus eine Verbindung zum RDBMS-Server hergestellt werden kann. Geben Sie den folgenden Befehl ein:

   ```
   catalog tcpip node
   Knoten remote Host \
   server Service
   ```

   Dabei gilt:

   Knoten
   Steht für den Namen des Systems, auf dem sich die

Client-Datenbank befindet. Dieser Name darf in der Knotenverzeichnisliste nur einmal vorkommen.

Host Steht für den Namen des RDBMS-Servers.

Service
Steht für den Server-Exemplareintrag. Der Server-Exemplareintrag befindet sich für UNIX in der Datei '/etc/services' und für Windows NT in der Datei '\winnt\system32\drivers\etc\services'. Ein Server-Exemplareintrag muss sowohl im RIM-Host als auch im RDBMS-Server vorliegen.

4. Prüfen Sie, ob der Knoten erstellt wurde. Geben Sie den folgenden Befehl ein:

   ```
   list node directory
   ```

   Daraufhin werden der Name, Host-Name und Servername des von Ihnen erstellten Knotens angezeigt.

5. Katalogisieren Sie die Serverdatenbank in dem Client-Exemplar auf dem RIM-Host. Gehen Sie dazu wie folgt vor:

   a. Geben Sie folgenden Befehl ein, um die Datenbank zu katalogisieren:

      ```
      catalog database Datenbank as Aliasname at node \
      Knoten
      ```

      Dabei gilt:

      Datenbank
      Steht für den Namen der Datenbank auf dem RDBMS-Server, die Sie für Tivoli Inventory erstellt haben.

      Aliasname
      Steht für den Aliasnamen des Client-Exemplars bzw. den auf dem Server verwendeten Datenbanknamen.

      Knoten
      Steht für den Namen des Systems, auf dem sich das von Ihnen erstellte Client-Exemplar befindet.

b. Prüfen Sie, ob die Datenbank erstellt wurde. Geben Sie den folgenden Befehl ein:

```
list database directory
```

Daraufhin werden der Aliasname, der Name der entsprechenden Datenbank und der Knoten, auf dem sich die Server-Datenbank befindet, angezeigt.

Informix konfigurieren

Vor der Installation von Tivoli Inventory müssen Sie zunächst das Verwaltungssystem für relationale Informix-Datenbanken installieren und ein Informix-Exemplar auf dem RDBMS-Server erstellen. Dabei müssen Sie die Datenbank als Benutzer 'Informix' erstellen. Anweisungen zum Erstellen eines Informix-Exemplars und einer Informix-Datenbank können Sie der Informix-Dokumentation entnehmen. Nach Ausführung der Skripts für die Installation des Tivoli Inventory-Schemas wird die von Ihnen erstellte Datenbank als Konfigurations-Repository von Tivoli Inventory verwendet. Weitere Informationen können Sie dem Abschnitt „Informix-Datenbank als Konfigurations-Repository erstellen" auf Seite 69 entnehmen.

Wählen Sie einen verwalteten Knoten als RIM-Host aus. Weitere Informationen zur Auswahl eines RIM-Hosts können Sie dem Abschnitt „RIM-Host auswählen" auf Seite 30 entnehmen.

Es wird empfohlen, den RDBMS-Server und den RIM-Host jeweils auf einem eigenen System zu installieren. Da eine Kommunikation zwischen dem RDBMS-Server und dem RIM-Host erforderlich ist, muss auf dem RIM-Host die Informix-Client-Software installiert sein.

MS SQL-Server konfigurieren

Installieren Sie den MS SQL-Server auf dem RDBMS-Server. Installationsanweisungen können Sie der Dokumentation zum MS SQL-Server entnehmen.

Achtung:
Der im Verlauf der Installation des MS SQL-Servers angezeigte Dialog 'Installationsoptionen' enthält die Option 'Sortierreihenfolge'. Für den Einsatz des MS SQL-Servers als Konfigurations-Repository von Tivoli Inventory müssen Sie eine Sortierreihenfolge nach Verzeichnis (unter Beachtung der Groß-/Kleinschreibung) angeben.

Wählen Sie einen verwalteten Windows NT-Knoten als RIM-Host aus. Weitere Informationen zur Auswahl des RIM-Hosts können Sie dem Abschnitt „RIM-Host auswählen" auf Seite 30 entnehmen. Nach der Installation von Tivoli Inventory können Sie die Skripts zum Erstellen der Datenbank und Installieren des Schemas für das Konfigurations-Repository ausführen. Weitere Informationen hierzu können Sie dem Abschnitt „MS SQL-Server als Konfigurations-Repository erstellen" auf Seite 72 entnehmen.

Es wird empfohlen, den RDBMS-Server und den RIM-Host auf unterschiedlichen Systemen zu installieren, daher müssen Sie den MQ SQL-Server oder -Client auf dem RIM-Host installieren.

Oracle konfigurieren

Installieren Sie das Verwaltungssystem für relationale Oracle-Datenbanken und SQL*Net auf dem RDBMS-Server. Anweisungen zur Installation können Sie der Oracle-Dokumentation entnehmen. Wählen Sie einen verwalteten Knoten als RIM-Host aus. Weitere Informationen können Sie dem Abschnitt „RIM-Host auswählen" auf Seite 30 entnehmen. Nach der Installation von Tivoli Inventory können Sie die Skripts zum Erstellen der Datenbank und Installieren des Schemas für das Konfigurations-Repository ausführen. Weitere Informationen hierzu können Sie dem Abschnitt „Oracle-Datenbank als Konfigurations-Repository erstellen" auf Seite 77 entnehmen.

Es wird empfohlen, für den RDBMS-Server und den RIM-Host unterschiedliche Systeme zu verwenden. Folgende Schritte sind zur Konfiguration des RIM-Hosts als Oracle-Client unter Verwendung von SQL*Net erforderlich (wenn Sie Oracle-Namen verwenden möchten, sehen Sie in der Dokumentation zu Oracle nach):

1. Installieren Sie die Oracle-Client-Software auf dem RIM-Host (dazu gehört auch SQL* Plus).
2. Kopieren Sie die Datei 'tnsnames.ora' von Verzeichnis $ORACLE_HOME/network/admin auf dem RDBMS-Server in das Verzeichnis $ORACLE_HOME/network/admin auf dem RIM-Host, dabei bezeichnet $ORACLE_HOME den Pfad für das Verzeichnis, in dem der Oracle-Server bzw. -Client installiert ist.
3. Prüfen Sie, ob die Datei 'tnsnames.ora' entsprechend Ihrer Konfiguration aktualisiert wurde. Dazu müssen Sie die Angaben für den Host-Namen auf dem Server, die ID des Oracle-Exemplars, den Anschluss, mit dem sqlnet verbunden ist, sowie das Übertragungsprotokoll auf ihre Richtigkeit überprüfen.

Sybase konfigurieren

Installieren Sie das Verwaltungssystem für relationale Sybase-Datenbanken. Anweisungen hierzu können Sie der Dokumentation zu Sybase entnehmen. Wählen Sie einen verwalteten Knoten als RIM-Host aus. Weitere Informationen können Sie dem Abschnitt „RIM-Host auswählen" auf Seite 30 entnehmen. Nach der Installation von Tivoli Inventory können Sie die Skripts zum Erstellen der Datenbank und Installieren des Schemas für das Konfigurations-Repository ausführen. Weitere Informationen hierzu können Sie dem Abschnitt „Sybase-Datenbank als Konfigurations-Repository erstellen" auf Seite 81 entnehmen.

Es wird empfohlen, für den RDBMS-Server und den RIM-Host unterschiedliche Systeme zu verwenden. Folgende Schritte sind zur Konfiguration eines RIM-Hosts als Client eines RDBMS-Servers erforderlich:

1. Installieren Sie die Sybase-Client-Software auf dem RIM-Host (dazu gehört auch isql).
2. Falls es sich bei dem RIM-Host um ein Windows NT-System handelt, müssen Sie sicherstellen, dass die Variable 'PATH' das Verzeichnis enthält, in dem die Sybase-DLLs installiert sind.
3. Kopieren Sie die Schnittstellendatei ('interfaces') vom RDBMS-Server in das Verzeichnis auf dem RIM-Host, in dem die Sybase-Client-Software installiert ist.

Wenn es sich bei dem RIM-Host um ein Windows NT-System handelt, müssen Sie sicherstellen, dass sich die Datei 'SQL.INI' im Verzeichnis $SYBASE/INI auf dem RIM-Host befindet; dabei bezeichnet $SYBASE den Pfad für das Verzeichnis, in dem der Sybase-Server bzw. -Client installiert ist.

Anmerkung: Wenn der RDBMS-Server auf einer Solaris-Maschine installiert ist, der RIM-Host jedoch nicht (oder umgekehrt), dürfen Sie die Schnittstellendatei nicht auf den RIM-Host kopieren. Die Schnittstellendatei für Solaris ist mit anderen Betriebssystemen nicht kompatibel. In diesem Fall müssen Sie eine neue, kompatible Schnittstellendatei erstellen. Wenden Sie sich dazu an Ihren Datenbankadministrator.

Tivoli Inventory-Produkte auf Zielsystemen installieren - Vorbereitungen

Nach Auswahl des RIM-Hosts sowie Installation und Konfiguration des RDBMS-Servers müssen Sie Tivoli Inventory Server oder Tivoli Inventory Gateway auf den entsprechenden Systemen in der TMR installieren.

Es wird empfohlen, vor und nach der Installation von Tivoli Inventory eine Sicherung der Tivoli Management Framework-Objektdatenbank für alle Systeme innerhalb der TMR vorzunehmen. Dadurch können Sie den Status, den die Objektdatenbank vor der Installation innehatte, wiederherstellen. Dies ist ratsam für den Fall, dass bei der Installation von Tivoli Inventory aus irgendwelchen Gründen Probleme auftreten. Wählen Sie auf der Tivoli-Arbeitsoberfläche aus dem Menü 'Arbeitsoberfläche' die Option **Sichern** aus, um den Tivoli-Server sowie die Tivoli-Clients zu sichern. Hierzu können Sie auch den Befehl **wbkupdb** verwenden. Weitere Informationen zum Befehl **wbkupdb** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

In der folgenden Tabelle sind die für die Installation von Tivoli Inventory erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Tivoli Inventory-Produkte installieren	Tivoli-Arbeitsoberfläche	**Install_product** oder **senior**

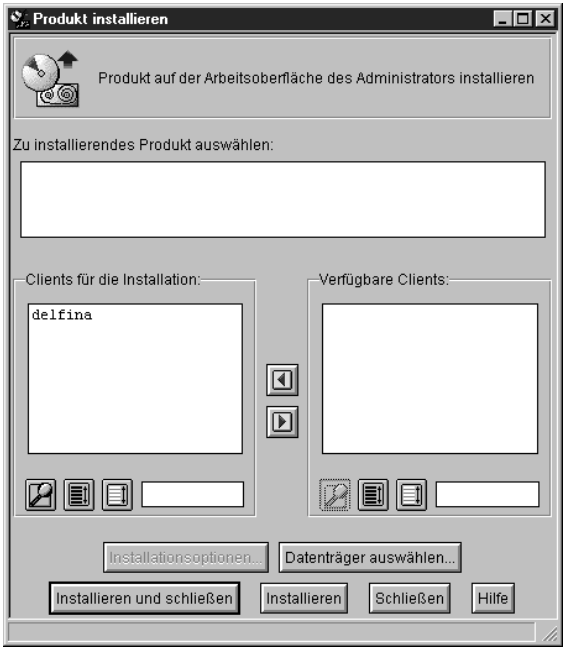
Sie können Tivoli Inventory Version 4.0 über die Tivoli-Arbeitsoberfläche sowie über die Befehlszeile installieren. Die Installation von Tivoli Inventory auf den Zielsystemen gliedert sich in die folgenden Schritte:

1. Datenträger auswählen
2. Zwischen Tivoli Inventory Server oder Tivoli Inventory Gateway auswählen
3. Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren
4. Tivoli Inventory Gateway installieren

Datenträger auswählen

Folgende Schritte sind erforderlich, um das Laufwerk auszuwählen, auf dem die CD-ROM mit der Tivoli-Software installiert ist, und um über die Tivoli-Arbeitsoberfläche auf die Tivoli Inventory-Dateien zuzugreifen, die für die Installation erforderlich sind:

1. Starten Sie die Tivoli-Arbeitsoberfläche auf dem TMR-Server.
2. Wählen Sie aus dem Menü 'Arbeitsoberfläche' nacheinander die Optionen **Installieren –> Produkt installieren** aus. Daraufhin wird der Dialog 'Produkt installieren' geöffnet.



Falls die Tivoli Inventory-Produkte in der Liste ’Zu installierendes Produkt auswählen’ bereits enthalten sind, gehen Sie zum Abschnitt „Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren” auf Seite 43. Fahren Sie andernfalls mit Schritt 3 fort.

3. Klicken Sie auf **Datenträger auswählen**. Daraufhin wird der Dialog ’Datei-Browser’ geöffnet.

4. Geben Sie im Feld ’Pfadname’ den Pfadnamen der CD-ROM mit der Tivoli-Software mit Hilfe einer der folgenden Methoden ein:

   - Geben Sie im Feld ’Pfadname’ den vollständigen Pfad ein.
   - Durchsuchen Sie das Dateisystem wie folgt:

     a. Wählen Sie in der Liste ’Hosts’ den Host (bzw. das Laufwerk) aus, auf dem sich die CD-ROM befindet. Nach der Auswahl eines Hosts wird die Liste ’Verzeichnisse’ aktualisiert und zeigt die Verzeichnisse (unter dem Stammverzeichnis) auf dem von Ihnen ausgewählten Host an.

     b. Doppelklicken Sie in der Liste ’Verzeichnisse’ auf das Verzeichnis, in dem sich der Installationsdatenträger befindet. Bei Auswahl eines Verzeichnisses wird die Liste ’Dateien’ aktualisiert.

     c. Klicken Sie auf **Pfad festlegen**.

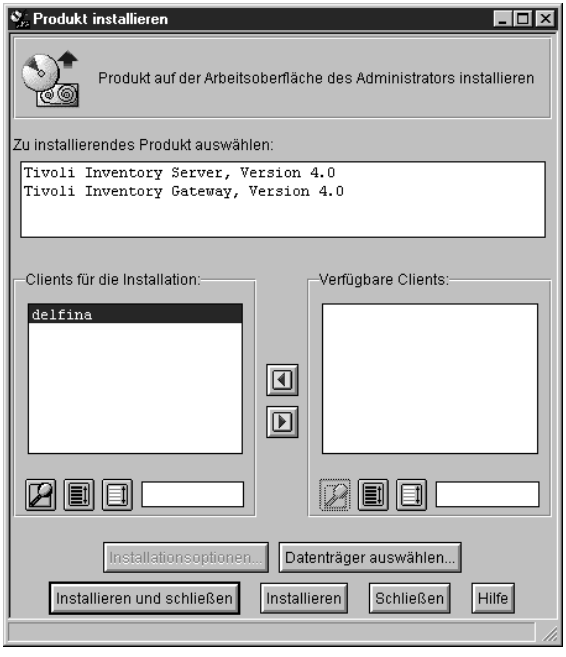
5. Klicken Sie auf **Datenträger festlegen und schließen**; daraufhin kehren Sie in den Dialog 'Produkt installieren' zurück. Die folgenden Produkte, die installiert werden können, werden in der Liste 'Zu installierendes Produkt auswählen' angezeigt:

   - Tivoli Inventory Server Version 4.0

     Enthält die Dateien, mit denen Sie Profile erstellen und verteilen sowie Tivoli Inventory-Befehle ausführen können. Installieren Sie dieses Produkt auf dem TMR-Server und auf den verwalteten Knoten, über die die Systeme in Ihrer TMR verwaltet werden sollen.

   - Tivoli Inventory Gateway Version 4.0

     Enthält die Dateien, die Gateways die Ausführung von Tivoli Inventory-Funktionen auf Endpunkten ermöglichen. Dieses Produkt muss auf allen verwalteten Knoten installiert werden, die Sie als Gateways konfiguriert haben.



Zwischen Tivoli Inventory Server oder Tivoli Inventory Gateway auswählen

Auf verwalteten Knoten können Sie sowohl Tivoli Inventory Server als auch Tivoli Inventory Gateway installieren. Sie können jedoch Zeit und Speicherplatz sparen, wenn Sie bei der Installation Folgendes berücksichtigen:

- Auf verwalteten Knoten, die für die Erstellung bzw. Verteilung von Inventory-Profilen und für die Ausführung von Tivoli Inventory-Befehlen verwendet werden, sollten Sie nur Tivoli Inventory Server installieren. Die hierzu erforderlichen Schritte sind im Abschnitt „Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren" aufgeführt.
- Auf Systemen, die als Endpunkt-Gateways verwendet werden, sollten Sie nur Tivoli Inventory Gateway installieren. Die hierzu erforderlichen Schritte sind im Abschnitt „Tivoli Inventory Gateway installieren" auf Seite 55 aufgeführt.
- Auf verwalteten Knoten, die für alle oben erwähnten Funktionen verwendet werden, sollten Sie Tivoli Inventory Server und Tivoli Inventory Gateway installieren.

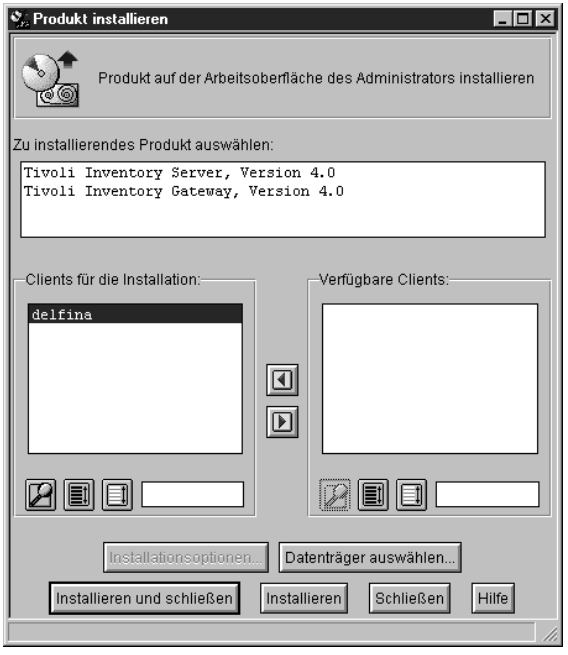
Weitere Informationen zur Installation von verwalteten Knoten und Gateways können Sie dem Handbuch *Tivoli Management Framework: Einsatzplanung* entnehmen.

Tivoli Inventory Server auf dem TMR-Server und auf verwalteten Knoten installieren

Nach Auswahl der Datenträger können Sie nun Tivoli Inventory Server Version 4.0 sowie die Informationen zu den Tivoli Inventory-Befehlen (MAN-Pages) auf dem TMR-Server und auf den verwalteten Knoten installieren, über die Profile erstellt und verteilt sowie Tivoli Inventory-Befehle ausgeführt werden sollen. Die Installation kann entweder über Tivoli-Arbeitsoberfläche oder die Befehlszeile erfolgen.

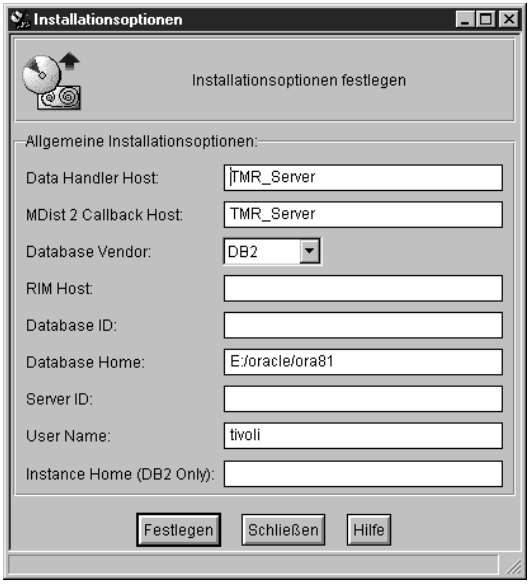
Inventory über die Arbeitsoberfläche installieren

1. Wählen Sie aus dem Menü 'Arbeitsoberfläche' nacheinander die Optionen **Installieren –> Produkt installieren** aus. Daraufhin wird der Dialog 'Produkt installieren' geöffnet.



2. Wählen Sie in der Liste 'Zu installierendes Produkt auswählen' im Dialog 'Produkt installieren' den Eintrag **Tivoli Inventory Server Version 4.0** aus.

   Daraufhin wird der Dialog 'Installationsoptionen' geöffnet. In diesem Dialog sind Angaben zum Verwaltungssystem für relationale Datenbanken sowie zum RIM-Host erforderlich.



3. Bearbeiten Sie den Dialog ’Installationsoptionen’. Im Folgenden finden Sie eine Beschreibung der erforderlichen Angaben. Informationen, die speziell Ihr Verwaltungssystem für relationale Datenbanken betreffen, finden Sie im entsprechenden Abschnitt am Ende dieser Beschreibung.

   a. Geben Sie im Feld ’Data Handler Host’ den Namen des verwalteten Knotens ein, der als Inventory Data Handler fungieren soll. Es wird empfohlen, dass Sie den Inventory Data Handler *nicht* auf dem verwalteten Knoten, der als TMR-Server fungiert, einrichten.

   b. Geben Sie im Feld ’MDist 2 Callback-Host’ den Namen des verwalteten Knotens ein, den Sie als TMR-Server konfiguriert haben. Mit diesem Schritt geben Sie den verwalteten Knoten an, auf dem das Callback-Objekt von Inventory installiert wird.

   c. Wählen Sie aus der Liste ’Database Vendor’ den Namen des Verwaltungssystems für relationale Datenbanken aus, das Sie für Tivoli Inventory einsetzen.

d. Geben Sie im Feld 'RIM Host' den Namen des verwalteten Knotens ein, den Sie als RIM-Host der TMR konfiguriert haben. Falls Sie noch keinen RIM-Host ausgewählt haben, lesen Sie den Abschnitt „RIM-Host auswählen" auf Seite 30.

e. Geben Sie im Feld 'Database ID' den Namen der Tivoli Inventory-Datenbank im Verwaltungssystem für relationale Datenbanken an.

f. Geben Sie im Feld 'Database Home' den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist.

g. Der Eintrag im Feld 'Server ID' hängt von dem Verwaltungssystem für relationale Datenbanken ab, das Sie für Tivoli Inventory verwenden. Informationen zu den Angaben, die Sie hier machen müssen, finden Sie im entsprechenden Abschnitt zum Verwaltungssystem für relationale Datenbanken am Ende dieser Beschreibung. Diese Angabe ermöglicht es dem Verwal-
tungssystem für relationale Datenbanken, eine Verbindung zu RIM herzustellen.

h. Geben Sie im Feld 'User Name' den Namen des Benutzers des Konfigurations-Repositorys von Tivoli Inventory ein.

i. Bei Verwendung von DB2 müssen Sie im Feld 'Instance Home' den Wert der Umgebungsvariable 'INSTHOME' übernehmen. Dabei handelt es sich um das Verzeichnis, in dem das Client-Exemplar für die DB2-Inventory-Datenbank erstellt wurde.

Die Installationsangaben, die für das von Ihnen verwendete Verwaltungssystem für relationale Datenbanken erforderlich sind, können Sie den folgenden Abschnitten entnehmen:

- **DB2**
  - Database ID - Geben Sie den Namen der DB2-Server-Datenbank ein, die Sie für Tivoli Inventory erstellt haben. Wurde ein ferner Client erstellt, der einen Aliasnamen verwendet, muss der Aliasname eingegeben werden.

- Database Home - Geben Sie den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist. Unter UNIX muss hier der Wert der Umgebungsvariable 'DB2DIR' übernommen werden.
- Server ID - Diese Angabe ermöglicht es dem Verwaltungssystem für relationale Datenbanken, eine Verbindung zum RIM herzustellen. Geben Sie **tcpip** ein.
- User Name - Geben Sie den Benutzernamen des Exemplareigners ein, der über eine Zugriffsberechtigung für die Tivoli Inventory-Datenbank verfügt. Weitere Informationen zu DB2 können Sie dem Abschnitt „Verwaltungssystem für relationale Datenbanken (RDBMS) konfigurieren" auf Seite 31 entnehmen.
- Instance Home - Übernehmen Sie in diesem Feld den Wert der Umgebungsvariable 'INSTHOME'. Dabei handelt es sich um das Verzeichnis, in dem das Client-Exemplar für die DB2-Datenbank von Tivoli Inventory erstellt wurde. In diesem Feld ist nur bei Verwendung einer DB2-Datenbank eine Angabe erforderlich.

■ **Informix**

- Database ID - Geben Sie den Namen der Informix-Server-Datenbank ein, die Sie für Tivoli Inventory erstellt haben. Wurde ein ferner Client erstellt, der einen Aliasnamen verwendet, muss der Aliasname eingegeben werden.
- Database Home - Geben Sie den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist. Unter UNIX muss hier der Wert der Umgebungsvariable '$INFORMIXDIR' übernommen werden.
- Server ID - Steht für den Namen des Servers. Dies ist der Wert, der über $INFORMIXDIR festgelegt wird.
- User Name - Geben Sie 'informix' ein.
- Instance Home - Keine Angabe erforderlich; dieses Feld ist nur für DB2 relevant.

- **MS SQL-Server**
  - Database ID - Geben Sie **inv_db** ein. Dabei handelt es sich um den Standardnamen der Datenbank, die wie in Kapitel „Konfigurations-Repository erstellen” auf Seite 63 beschrieben, bei Ausführung des Skripts ’inv_ms_sql_admin.sql’ oder ’inv_syb_admin.sql’ erstellt wurde.
  - Database Home - Geben Sie den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist.
  - Server ID - Diese Angabe ermöglicht es dem Verwaltungssystem für relationale Datenbanken, eine Verbindung zum RIM herzustellen. Geben Sie den Namen des Servers ein; dabei handelt es sich um das System, auf dem der MS SQL-Server installiert ist.
  - User Name - Geben Sie den Namen des Benutzers ein, der der Eigner des Konfigurations-Repositorys von Tivoli Inventory ist. Der Standardwert ist ’invtiv’.
  - Instance Home - Keine Angabe erforderlich; dieses Feld ist nur für DB2 relevant.
- **Oracle**
  - Database ID - Übernehmen Sie in diesem Feld den Wert der Umgebungsvariable ’ORACLE_SID’. Bei diesem Wert handelt es sich um die ID des Oracle-Exemplars; diese ist in der Datei ’tnsnames.ora’ im Verzeichnis $ORACLE_HOME/network/admin enthalten. Für Windows NT und UNIX wird bei der Installation von Oracle der Standardwert ’ORCL’ gesetzt.
  - Database Home - Geben Sie den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist. Hier wird der Wert von ’ORACLE_HOME’ übernommen, der das Verzeichnis angibt, in dem die Oracle-RDBMS-Software installiert ist.
  - Server ID - Diese Angabe ermöglicht es dem Verwaltungssystem für relationale Datenbanken, eine Verbindung zum RIM herzustellen. Übernehmen Sie in diesem Feld

den Wert der Umgebungsvariable 'TWO_TASK'. Dieser Wert ist in der Datei 'tnsnames.ora' im Verzeichnis $ORACLE_HOME/network/admin enthalten.

- User Name - Geben Sie den Namen des Benutzers ein, der der Eigner des Konfigurations-Repositorys von Tivoli Inventory ist. Falls es sich bei dem RIM-Host nicht um ein Windows NT-System handelt, ist der Standardwert 'invtiv'.

  Handelt es sich um ein Windows NT-System, müssen Sie **invtiv@***TWO_TASK* eingeben; dabei steht *TWO_TASK* für den Wert der Umgebungsvariablen 'TWO_TASK'.

- Instance Home - Keine Angabe erforderlich; dieses Feld ist nur für DB2 relevant.

■ **Sybase**

- Database ID - Geben Sie **inv_db** ein. Dabei handelt es sich um den Standardnamen der Datenbank, die wie in Kapitel „Konfigurations-Repository erstellen" auf Seite 63 beschrieben, bei Ausführung des Skripts 'inv_ms_sql_admin.sql' oder 'inv_syb_admin.sql' erstellt wurde.

- Database Home - Geben Sie den Pfad für das Verzeichnis ein, in dem die RDBMS-Server- bzw. -Client-Software auf dem RIM-Host installiert ist. Hier wird der Wert der Umgebungsvariable 'SYBASE' übernommen, die das Verzeichnis angibt, in dem die Sybase-RDBMS-Software installiert ist. Befindet sich der RIM-Host nicht auf demselben System wie der RDBMS-Server, müssen Sie den Wert von 'SYBASE' übernehmen; dieser gibt das Verzeichnis auf dem RIM-Host an, das die Schnittstellendatei enthält.

- Server ID - Übernehmen Sie in diesem Feld den Wert der Umgebungsvariable 'DSQUERY'. Diese finden Sie in der Schnittstellendatei.

- User Name - Geben Sie den Namen des Benutzers ein, der der Eigner des Konfigurations-Repositorys von Tivoli Inventory ist. Der Standardwert ist 'invtiv'.

- Instance Home - Keine Angabe erforderlich; dieses Feld ist nur für DB2 relevant.

4. Klicken Sie auf **Festlegen**, um die Angaben im Dialog 'Installationsoptionen' zu sichern und in den Dialog 'Produkt installieren' zurückzukehren.
5. Wählen Sie in der Liste 'Clients für die Installation' alle verwalteten Knoten aus, auf denen Sie Tivoli Inventory *nicht* installieren möchten (es werden nur verwaltete Knoten angezeigt); klicken Sie dann auf den Rechtspfeil, um die ausgewählten verwalteten Knoten in die Liste 'Verfügbare Clients' zu verschieben.
6. Starten Sie die Installation von Tivoli Inventory Server, indem Sie auf **Installieren** klicken. Im Dialog 'Produkt installieren' werden die Vorgänge aufgeführt, die bei der Installation aufgelistet werden; außerdem enthält er Hinweise auf Fehler, die Sie vor der Installation beheben sollten. Sie können eine der folgenden Optionen auswählen:
   - Überprüfen Sie die Statusinformationen, und klicken Sie auf **Installation fortsetzen**. Sobald die Installation beendet ist, wird im Dialog 'Produkt installieren' eine entsprechende Nachricht angezeigt.
   - Um Tivoli Inventory Server zu einem späteren Zeitpunkt zu installieren, klicken Sie auf **Abbrechen**. Ihre Einstellungen werden nicht gesichert.
7. Aktualisieren Sie die Subskriptionen für die Hinweisgruppen aller Administratoren, die Tivoli Inventory-Hinweise erhalten sollen. Anweisungen zur Änderung der Administratorsubskriptionen für Hinweisgruppen und anderer Merkmale können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.
8. Fügen Sie für jeden Richtlinienbereich, in dem ein Inventory-Profil erstellt werden soll, die verwaltete Ressourcenart 'InventoryConfig' hinzu. Anweisungen zum Hinzufügen verwalteter Ressourcenarten in Richtlinienbereichen können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

9. Überprüfen Sie die Tivoli Inventory-Hinweisgruppe, um sicherzustellen, dass das RIM-Objekt erstellt wurde. Ist bei der Erstellung des RIM-Objekts ein Fehler aufgetreten, müssen Sie mit dem Befehl **wcrtrim** das RIM-Objekt für den RIM-Host erstellen. Weitere Informationen zum Befehl **wcrtrim** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Kennwörter für RIM-Objekte ändern

Bei der Installation von Tivoli Inventory werden folgende RIM-Objekte erstellt:

invdh_1
: Der Inventory Data Handler schreibt mit Hilfe dieses RIM-Objekts Suchlaufdaten in das Konfigurations-Repository.

inv_query
: Über dieses RIM-Objekt werden Abfragen an das Konfigurations-Repository abgesetzt. Dieses RIM-Objekt wird von den Befehlen **winvfilter**, **winvrmnode** und **winvsig** und der grafischen Benutzerschnittstelle (GUI) von Tivoli Inventory verwendet.

Darüber hinaus wird eines der folgenden RIM-Objekte erstellt:

inv_40
: Diese RIM-Objekt wird nur erstellt, wenn Tivoli Inventory Version 4.0 zusammen mit Tivoli Inventory Version 3.6.2 in derselben TMR installiert ist. Dieses RIM-Objekt wird ausschließlich von Tivoli Software Distribution für den Zugriff auf das Konfigurations-Repository verwendet.

inventory
: Dieses RIM-Objekt wird nur erstellt, wenn Tivoli Inventory Version 4.0 und Tivoli Inventory Version 3.6.2 nicht in derselben TMR installiert sind. Dieses RIM-Objekt wird ausschließlich von Tivoli Software Distribution für den Zugriff auf das Konfigurations-Repository verwendet.

Standardmäßig werden alle RIM-Objekte auf demselben verwalteten Knoten erstellt.

Die RIM-Objekte 'invdh_1' und 'inv_query' verwenden beide das Konfigurations-Repository von Tivoli Inventory Version 4.0. Benutzer von Software Distribution können die RIM-Objekte 'inventory' und 'inv_40' so konfigurieren, dass entweder das Konfigurations-Repository von Tivoli Inventory Version 4.0 oder das Konfigurations-Repository von Tivoli Inventory Version 3.6.2 verwendet wird. Weitere Informationen zu den RIM-Objekten 'inv_40' und 'inventory' und deren Konfiguration können Sie den *Tivoli Inventory Release Notes Version 4.0* entnehmen.

Das Standardkennwort für alle RIM-Objekte lautet 'tivoli'. Das Kennwort für das jeweilige RIM-Objekt sollte jedoch mit dem Kennwort des Konfigurations-Repositorys, auf das es zugreift, übereinstimmen. Daher müssen Sie die Kennwörter aller RIM-Objekte ändern, so dass diese den Kennwörtern für das zugehörige Konfigurations-Repository entsprechen. Darüber hinaus müssen Sie die Kennwörter für die RIM-Objekte ändern, wenn Sie das Kennwort für das zugehörige Konfigurations-Repository ändern.

Mit dem Befehl **wsetrimpw** können Sie das Kennwort für ein RIM-Objekt ändern. Verwenden Sie beispielsweise folgende Befehle, um das Kennwort für die RIM-Objekte 'invdh_1', 'inv_query' und 'inventory' zu ändern:

```
wsetrimpw invdh_1 altes_Kennwort neues_Kennwort
wsetrimpw inv_query altes_Kennwort neues_Kennwort
wsetrimpw inventory altes_Kennwort neues_Kennwort
```

Dabei steht *altes_Kennwort* für das aktuelle Kennwort und *neues_Kennwort* für das gewünschte neue Kennwort. Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Über die Befehlszeile

Im folgenden Beispiel wird mit dem Befehl **winstall** Tivoli Inventory Server auf dem verwalteten Knoten 'oak' von einem CD-ROM-Image installiert, die Dateien werden in das Standardverzeichnis gestellt und eine Sybase-RDBMS-Datenbank wird als Konfigurations-Repository eingerichtet. Dieses Beispiel gilt für alle unterstützten RDBMS-Datenbanken außer DB2-Datenbanken.

```
winstall –c /cdrom -i 40_Inv @INV_DATA_HOST@=oak \
@INV_CB_HOST@=oak @RDBMS_Vendor@=Sybase \
@RDBMS_Host@=olympus @RDBMS_DB_Name@=inv_db \
@RDBMS_DB_Home@=/opt/sybase @RDBMS_DB_Param_one@=INV \
oak @RDBMS_DB_UserName@=invtiv
```

Dabei gilt:

–c /cdrom
Gibt /cdrom als Pfad des CD-ROM-Abbilds an.

–i 40_Inv
Gibt die Produktindexdatei 'INVENTOR' an, von der Tivoli Inventory installiert wird.

@INV_DATA_HOST@=oak
Gibt an, dass es sich bei dem verwalteten Knoten 'oak' um den Host für den Inventory Data Handler handelt.

@INV_CB_HOST@=oak
Gibt an, dass es sich bei dem verwalteten Knoten 'oak' um den MDist 2-Callback-Host handelt.

@RDBMS_Vendor@=Sybase
Gibt an, dass eine Sybase-RDBMS-Datenbank für Tivoli Inventory verwendet wird.

@RDBMS_Host@=olympus
Gibt an, dass das System 'olympus' als RIM-Host eingesetzt wird.

@RDBMS_DB_Name@=inv_db
Gibt als Datenbank-ID den Namen 'inv_db' an. Dabei handelt es sich um den Standardnamen der Datenbank, die bei Ausführung des Skripts 'inv_syb_admin.sql' erstellt wurde.

@RDBMS_DB_Home@=/opt/sybase
Gibt als Datenbankposition das Verzeichnis /opt/sybase an; in diesem Verzeichnis befinden sich 'isql' und die Schnittstellendatei.

@RDBMS_DB_Param_one@=INV
Gibt an, dass als Server-ID der Wert 'INV' übernommen wird. Hierbei handelt es sich um den Wert der Umgebungsvariable 'DSQUERY', die sich in der Schnittstellendatei befindet.

oak Gibt an, dass Tivoli Inventory Server auf dem verwalteten Knoten 'oak' installiert wird. Wenn Sie keinen verwalteten Knoten angeben, wird das Produkt auf allen verfügbaren Clients installiert.

@RDBMS_DB_UserName@=invtiv
Gibt 'invtiv' als RDBMS-Benutzername für das Konfigurations-Repository von Tivoli Inventory an.

Anmerkung: Das oben angeführte Beispiel für eine Installation über die Befehlszeile kann mit Ausnahme von DB2 für alle Verwaltungssysteme für relationale Datenbanken verwendet werden, die von Tivoli Inventory unterstützt werden. Bei Einsatz von DB2 muss in dem Befehl noch der folgende Aufruf angegeben werden:

`@RDBMS_DB_Param_two@=`*`Exemplarposition`*

Dabei gilt:

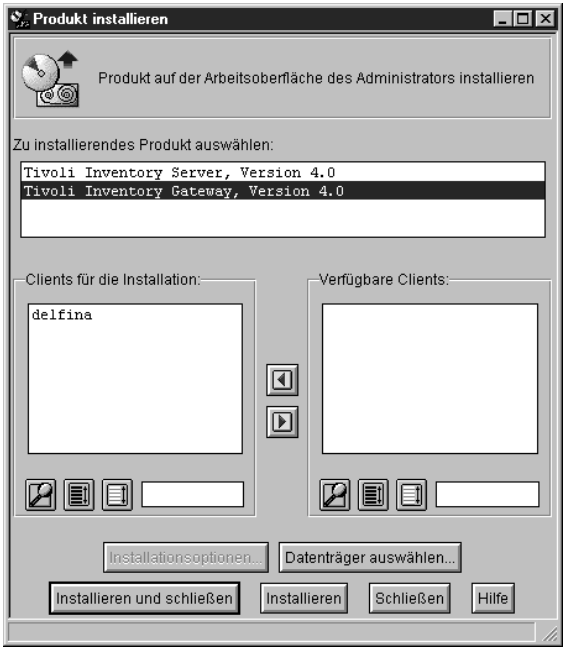
@RDBMS_DB_Param_two@=*Exemplarposition*
Dabei handelt es sich um das Verzeichnis, in dem das Client-Exemplar für die DB2-Datenbank von Tivoli Inventory erstellt wurde.

Weitere Informationen zum Befehl **winstall** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Tivoli Inventory Gateway installieren

Nach der Installation von Tivoli Inventory Server auf dem TMR-Server und den ausgewählten verwalteten Knoten können Sie Tivoli Inventory Gateway auf den verwalteten Knoten installieren, die als Gateways für Verwaltungsendpunkte fungieren sollen; gehen Sie dazu wie folgt vor:

1. Wählen Sie aus dem Menü 'Arbeitsoberfläche' nacheinander die Optionen **Installieren –> Produkt installieren** aus. Daraufhin wird der Dialog 'Produkt installieren' geöffnet.
2. Wählen Sie in der Liste 'Zu installierendes Produkt auswählen' im Dialog 'Produkt installieren' den Eintrag **Tivoli Inventory Gateway Version 4.0** aus.



Bei Auswahl von **Tivoli Inventory Gateway Version 4.0** werden die Dateien installiert, die Gateways die Ausführung von Tivoli Inventory-Funktionen auf Endpunkten ermöglichen. Dieses Produkt muss auf allen verwalteten Knoten installiert werden, die Sie als Gateways konfiguriert haben.

3. Wählen Sie in der Liste 'Clients für die Installation' alle verwalteten Knoten aus, auf denen Sie Tivoli Inventory Gateway *nicht* installieren möchten. Es werden nur verwaltete Knoten angezeigt.
4. Klicken Sie auf den Rechtspfeil, um die ausgewählten verwalteten Knoten in die Liste 'Verfügbare Clients' zu verschieben.
5. Starten Sie die Installation von Tivoli Inventory Gateway, indem Sie auf **Installieren** klicken. Im Dialog 'Produkt installieren' werden die Vorgänge aufgelistet, die bei der Installation ausgeführt werden; außerdem enthält er Hinweise auf Fehler, die Sie vor der Installation beheben sollten. Sie können eine der folgenden Optionen auswählen:
   - Überprüfen Sie die Statusinformationen, und klicken Sie auf **Installation fortsetzen**. Sobald die Installation beendet ist, wird im Dialog 'Produkt installieren' eine entsprechende Nachricht angezeigt.
   - Um Tivoli Inventory Gateway zu einem späteren Zeitpunkt zu installieren, klicken Sie auf **Abbrechen**. Ihre Einstellungen werden nicht gesichert.

Tivoli Inventory deinstallieren

Die Tivoli Inventory-Produkte der Version 4.0 können mit einem der folgenden Verfahren vom TMR-Server oder von einem Knoten in Ihrer TMR-Umgebung entfernt werden:

- Mit einem automatischen Deinstallationsprogramm, über das produktspezifische Deinstallationsskripts ausgeführt werden können
- Mit manuellen Deinstallationsskripts, mit denen Tivoli Inventory Server oder Tivoli Inventory Gateway gezielt entfernt werden können

Tivoli Inventory mit dem automatischen Deinstallationsprogramm entfernen

In Tivoli Management Framework steht ein Befehlszeilendienstprogramm zur Verfügung, mit dem Tivoli Management Framework-Anwendungen von einem angegebenen Knoten oder aus der ganzen TMR entfernt werden können. Bei dem Befehl **wuninst** handelt es sich um ein Oberflächenskript, von dem produktspezifische Deinstallationsskripts ausgeführt werden.

Mit dem Befehl **wuninst** können Sie Tivoli Inventory-Produkte von jedem System in Ihrer Umgebung oder aus der TMR entfernen. Weitere Informationen zur Befehlszeilensyntax und zur Verwendung des Befehls **wuninst** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Hinweise zur Verwendung und Syntax des Befehls für Tivoli Inventory erhalten Sie, wenn Sie in der Befehlszeile Folgendes eingeben:

```
wuninst
```

Das folgende Beispiel zeigt, wie Tivoli Inventory Server aus der gesamten TMR, deren TMR-Server 'pescado' heißt, entfernt wird. Dabei sollten Sie beachten, dass Tivoli Inventory Server vom TMR-Server und von allen verwalteten Knoten in der TMR, auf denen Tivoli Inventory Server installiert ist, entfernt wird, wenn es sich bei dem angegebenen Knoten um den TMR-Server handelt. Geben Sie den folgenden Befehl ein:

```
wuninst InventoryServer pescado -rmfiles
```

Dabei gilt:

InventoryServer
: Gibt das registrierte Produktkennzeichen für Tivoli Inventory Server an.

pescado
: Gibt den TMR-Server an. Da es sich bei ’pescado’ um den TMR-Server handelt, wird Tivoli Inventory Server von allen Knoten in der TMR entfernt.

–rmfiles
: Gibt an, dass alle lokalen Tivoli Management Framework-Datenbankobjekte sowie alle zugehörigen Dateien entfernt werden, selbst wenn es sich um gemeinsam benutzte Dateien handeln sollte.

Ohne Angabe der Option**–rmfiles** werden mit diesem Befehl nur die Tivoli Management Framework-Datenbankobjekte für den angegebenen Knoten entfernt. Handelt es sich bei dem angegebenen Knoten um den TMR-Server, werden bei Angabe dieser Option alle Datenbankobjekte entfernt.

Das folgende Beispiel zeigt, wie Tivoli Inventory Gateway aus der gesamten TMR, deren TMR-Server ’pescado’ heißt, entfernt wird. Geben Sie den folgenden Befehl ein:

```
wuninst InventoryGateway pescado -rmfiles -all
```

Dabei gilt:

InventoryGateway
Gibt das registrierte Produktkennzeichen für Tivoli Inventory Gateway an.

–rmfiles
Gibt an, dass alle lokalen Tivoli-Datenbankobjekte sowie alle zugehörigen Dateien entfernt werden, selbst wenn es sich um gemeinsam benutzte Dateien handelt.

–all Gibt an, dass Tivoli Inventory Gateway von allen Systemen in der TMR entfernt werden soll.

pescado
Gibt den TMR-Server an. Da es sich bei 'pescado' um den TMR-Server handelt, wird Tivoli Inventory Gateway von allen Knoten in der TMR entfernt.

Das folgende Beispiel zeigt, wie Datenbankobjekte von einem einzelnen Knoten mit dem Namen 'odin' entfernt werden. Geben Sie in der Befehlszeile den folgenden Befehl ein:

```
wuninst InventoryServer odin
```

Dabei gilt:

InventoryServer
Gibt das registrierte Produktkennzeichen für Tivoli Inventory Server an.

odin Gibt den Knoten an, von dem das Produkt entfernt wird.

Systeme aus dem Konfigurations-Repository entfernen

Wurde Tivoli Inventory von einem System in Ihrer Tivoli-Umgebung entfernt, ist es unter Umständen ratsam, die Einträge zu dem betreffenden System aus dem Konfiguration-Repository zu löschen. Dazu können Sie den Befehl **winvrmnode** verwenden. Weitere Informationen zu diesem Befehl finden Sie im Anhang „Befehle” auf Seite 239.

TEC-Server angeben

Wenn Sie Tivoli Inventory-Ereignisse an einen oder mehrere von Ihnen angegebene TEC-Server (Tivoli Enterprise Console) senden möchten, müssen Sie eine Tivoli Inventory-Regelbasis erstellen und die BAROC-Dateien in die TEC importieren. Die Tivoli Inventory-Anwendung sendet die Ereignisse an die TEC-Server in der Reihenfolge, in der Sie angegeben wurden. Sie müssen dieses Verfahren nicht verwenden, wenn Sie keine Ereignisse an TEC-Server senden möchten.

Weitere Informationen zum gemeinsamen Einsatz von Tivoli Inventory und TEC-Servern können Sie dem Abschnitt „wsetinvglobal” auf Seite 325 entnehmen.

Führen Sie die folgenden Schritte aus, um die BAROC-Dateien in die TEC zu importieren:

1. Erstellen Sie eine neue Regelbasis mit dem Namen ’Inventory’.
2. Kopieren Sie die Standardkategorien aus der TEC-Regelbasis in die neue Regelbasis mit dem Namen ’Inventory’.
3. Importieren Sie die BAROC-Dateien von Tivoli Inventory.
4. Kompilieren Sie die Inventory-Regelbasis.
5. Laden Sie die Inventory-Regelbasis in den TEC-Server.
6. Stoppen Sie den TEC-Server.
7. Starten Sie den TEC-Server erneut.

Weitere Informationen zu diesem Verfahren können Sie dem Handbuch *Tivoli Enterprise Console Rule Builder's Guide* entnehmen. Das folgende Beispiel zeigt, wie eine neue Regelbasis mit dem Namen Inventory erstellt und die BAROC-Datei für Tivoli Enterprise Console Version 3.6.*x* oder eine frühere Version importiert wird.

Anmerkung: Das folgende Verfahren kann nicht für höhere Versionen von Tivoli Enterprise Console angewandt werden.

1. Erstellen Sie eine neue Regelbasis mit dem Namen 'Inventory'.
   ```
   wcrtrb -d Host:$BINDIR/TME/TEC/rules/Inventory Inventory
   ```
2. Kopieren Sie die Standardkategorien aus der TEC-Regelbasis in die neue Regelbasis mit dem Namen 'Inventory'.
   ```
   wcprb Default Inventory
   ```
3. Importieren Sie die BAROC-Dateien von Tivoli Inventory.
   ```
   wimprbclass $BINDIR/TME/INVENTORY/tecad_inv.baroc Inventory
   ```
4. Kompilieren Sie die Inventory-Regelbasis.
   ```
   wcomprules Inventory
   ```
5. Laden Sie die Inventory-Regelbasis in den TEC-Server.
   ```
   wloadrb Inventory
   ```
6. Stoppen Sie den TEC-Server.
   ```
   wstopesvr
   ```
7. Starten Sie den TEC-Server erneut.
   ```
   wstartesvr
   ```

Weitere Vorgehensweise

Nach Installation der Tivoli Inventory-Produkte auf den Systemen in Ihrer Tivoli-Umgebung können Sie mit dem Kapitel „Konfigurations-Repository erstellen" auf Seite 63 fortfahren; dieses Kapitel enthält Informationen zum Erstellen des Konfigurations-Repositorys von Tivoli Inventory und zum Installieren der mit Tivoli Inventory Version 4.0 gelieferten Softwaresignaturen und -abfragen.

3

Konfigurations-Repository erstellen

Damit die Ergebnisse von Inventory-Suchläufen in einer RDBMS-Datenbank gespeichert werden können, müssen Sie die Tivoli Inventory-Datenbank, das so genannte Konfigurations-Repository, erstellen. Für den Zugriff auf die durch Suchläufe ermittelten Daten ist die Installation der mit Tivoli Inventory gelieferten Softwaresignaturen und Abfragen erforderlich. In diesem Kapitel werden die einzelnen Schritte erläutert, die bei der Erstellung des Schemas des Konfigurations-Repositorys von Tivoli Inventory, bei der Installation von Softwaresignaturen im Konfigurations-Repository sowie bei der Installation der Abfragen, über die Daten aus dem Konfigurations-Repository abgerufen werden, erforderlich sind.

Vor der Erstellung des Konfigurations-Repositorys müssen Sie zunächst einen RIM-Host auswählen, eine RDBMS-Datenbank installieren und konfigurieren sowie Tivoli Inventory in der TMR installieren. Falls dies noch nicht geschehen ist, lesen Sie zuerst in Kapitel „Tivoli Inventory Version 4.0 installieren” auf Seite 19 nach.

Bei Einsatz von Oracle, Sybase oder eines MS SQL-Servers stehen in Tivoli Inventory Skripts zur Verfügung, mit denen die Datenbank erstellt und das Schema des Konfigurations-Repositorys installiert werden kann. Bei der Verwendung von DB2 oder Informix müssen Sie die Datenbank manuell wie im Abschnitt „DB2 konfigurieren” auf Seite 32 bzw. „Informix konfigurieren” auf Seite 35 beschrieben

erstellen. Anschließend können Sie mit den für DB2 oder Informix zur Verfügung gestellten Skripts das Konfigurations-Repository-Schema erstellen.

Führen Sie nach der Konfiguration Ihrer Datenbank folgende Schritte aus:

1. Erstellen Sie unter Verwendung eines der folgenden Verfahren ein datenbankspezfisches Konfigurations-Repository:
   - DB2-Datenbank als Konfigurations-Repository erstellen
   - Informix-Datenbank als Konfigurations-Repository erstellen
   - MS SQL-Server als Konfigurations-Repository erstellen
   - Oracle-Datenbank als Konfigurations-Repository erstellen
   - Sybase-Datenbank als Konfigurations-Repository erstellen
2. Installieren Sie die Softwaresignaturen. Siehe hierzu „Softwaresignaturen installieren” auf Seite 85.
3. Erstellen Sie die Tivoli Inventory- und Subskriptionsabfragen. Siehe hierzu „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87.

DB2-Datenbank als Konfigurations-Repository erstellen

In diesem Abschnitt werden die einzelnen Schritte erläutert, die bei der Erstellung des Konfigurations-Repositorys von Tivoli Inventory Version 4.0 auf der Basis einer DB2-Datenbank erforderlich sind. Vor Ausführung dieser Skripts müssen Sie jedoch zunächst sicherstellen, dass die Umgebungsvariable DB2INSTANCE gesetzt ist.

Führen Sie vor Erstellung des Konfigurations-Repositorys zunächst Folgendes aus:

- Falls sich das Skript ’db2profile’ (oder ’db2cshrc’, wenn Sie die C-Shell verwenden) nicht im Benutzerverzeichnis des Exemplareigners befindet, das die Einstellungen für die Umgebungs-

variable DB2INSTANCE enthält, müssen Sie diese Umgebungsvariable auf den Anmeldenamen des Exemplareigners setzen.

- Stellen Sie sicher, dass der Pfad für DB2 in der Umgebung gesetzt ist, in der die Skripts ausgeführt werden.

Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen

Folgende Schritte sind zur Ausführung des Skripts 'inv_db2_admin.sql' erforderlich, das den für das Konfigurations-Repository erforderlichen Tabellenbereich sowie den temporären Tabellenbereich für den Exemplareigner des Konfigurations-Repositorys von Tivoli Inventory erstellt.

Das Skript 'inv_db2_admin.sql' erstellt 12.500 Seiten für einen Tabellenbereich; das entspricht 50 MB. Für jedes System, zu dem Informationen in der Datenbank gespeichert werden, sollten 125 Seiten bzw. 500 KB zur Verfügung gestellt werden. Für ein Konfigurations-Repository mit Informationen zu 1000 Knoten ist daher ein Platzbedarf von 125.000 Seiten bzw. 500 MB erforderlich. Zum Erstellen größerer Tabellenbereiche muss das Skript 'inv_db2_admin.sql' entsprechend angepasst werden.

Der Name sowie die Größe der Datendatei für den Tabellenbereich kann geändert werden, indem Sie die folgende Zeile wie folgt ändern:

```
USING (FILE 'inv_data_1' 12500)
```

in:

```
USING (FILE 'Datendatei' Größe_der_Datendatei)
```

Dabei steht *Datendatei* für den Namen, der der Datendatei zugewiesen werden soll, und *Größe_der_Datendatei* für die gewünschte Größe.

Folgende Schritte sind zur Erstellung des für das Konfigurations-Repository von Tivoli Inventory erforderlichen Tabellenbereichs erforderlich:

1. Kopieren Sie das Skript 'inv_db2_admin.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis, in dem DB2 auf dem DB2-Server oder dem RIM-Host ausgeführt werden kann.

2. Stellen Sie vor Ausführung des Skripts zunächst eine Verbindung zur DB2-Server-Datenbank her. Geben Sie dazu den folgenden Befehl in der Befehlszeile ein:

   ```
   db2 connect to Datenbank user Benutzer using Kennwort
   ```

   Dabei gilt:

   Datenbank
   : Steht für den Namen bzw. Aliasnamen der Tivoli Inventory-Datenbank.

   Benutzer
   : Steht für den Namen des Benutzers, der Eigner der Tivoli Inventory-Datenbank ist. Dabei handelt es sich um den Anmeldenamen des DB2-Exemplareigners.

   Kennwort
   : Steht für das Kennwort des über *Benutzer* angegebenen Benutzers.

3. Führen Sie das Skript 'inv_db2_admin.sql' über die Befehlszeile aus. Geben Sie den folgenden Befehl ein:

   ```
   db2 -f inv_db2_admin.sql -o -t -z \
   inv_db2_admin.log
   ```

   Bei Eingabe dieses Befehls wird das Skript ausgeführt, das den vom Konfigurations-Repository verwendeten Tabellenbereich erstellt. Darüber hinaus wird mit diesem Befehl die Ausgabe an den Bildschirm übertragen, das Semikolon als Begrenzer am Ende einer Anweisung angegeben und die Ausgabe in der Protokolldatei 'inv_db2_admin.log' aufgezeichnet. In der Protokolldatei wird angegeben, ob die Ausführung einer SQL-Anweisung erfolgreich verlaufen ist oder nicht.

4. Ersetzen Sie das Kennwort des Exemplareigners (standardmäßig 'tivoli') durch das Kennwort des Exemplareigners des Konfigurations-Repositorys. Mit dem Befehl **wsetrimpw** können Sie das

Standardkennwort 'tivoli' durch das Kennwort des Exemplareigners ersetzen. Bei dem Exemplareigner handelt es sich um den Benutzer, den Sie bei Erstellung der DB2-Datenbank und des DB2-Datenbankexemplars angegeben haben. Der Exemplareigner wurde auch als Benutzername im Feld 'User ID' im Dialog 'Installationsoptionen' angegeben.

Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Konfigurations-Repository-Schema installieren

Folgende Schritte sind zur Ausführung des Skripts 'inv_db2_schema.sql' erforderlich, mit dem das Schema des Konfigurations-Repositorys installiert wird. Dieses Schema definiert die Tabellen und Ansichten im Konfigurations-Repository. Folgende Schritte sind erforderlich:

1. Kopieren Sie das Skript 'inv_db2_schema.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis auf einem System mit der DB2 Client-Software.

2. Stellen Sie vor Ausführung des Skripts zunächst eine Verbindung zur DB2-Server-Datenbank her. Geben Sie dazu den folgenden Befehl in der Befehlszeile ein:

   ```
   db2 connect to Datenbank user Benutzer \
   using Kennwort
   ```

   Dabei gilt:

   Datenbank
   : Steht für den Namen bzw. Aliasnamen der Tivoli Inventory-Datenbank.

   Benutzer
   : Steht für den Namen des Benutzers, der Eigner der Tivoli Inventory-Datenbank ist.

   Kennwort
   : Steht für das Kennwort des über *Benutzer* angegebenen Benutzers.

3. Führen Sie das Skript 'inv_db2_schema.sql' über die Befehlszeile aus. Geben Sie den folgenden Befehl ein:

   ```
   db2 -f inv_db2_schema.sql -o -t \
   -z inv_db2_schema.log
   ```

   Bei Eingabe dieses Befehls wird das Skript ausgeführt, mit dem das Konfigurations-Repository-Schema installiert wird. Darüber hinaus wird mit diesem Befehl die Ausgabe an den Bildschirm übertragen, das Semikolon als Begrenzer am Ende einer Anweisung angegeben und die Ausgabe in der Protokolldatei 'inv_db2_schema.log' aufgezeichnet. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht.

4. Wenn Sie Protokolltabellen verwenden, wiederholen Sie die Schritte 1 bis 3 mit dem Skript 'h_inv_db2_schema.sql' und der zugehörigen Protokolldatei 'h_inv_db2_schema.log'.

Konfiguration prüfen

Folgende Schritte sind erforderlich, um die Verbindung zwischen dem RIM-Host und dem RDBMS-Server zu überprüfen und um festzustellen, ob das Konfigurations-Repository erstellt wurde:

1. Starten Sie auf dem RIM-Host eine interaktive SQL-Sitzung, indem Sie den folgenden Befehl eingeben:

   ```
   db2
   ```

2. Prüfen Sie, ob der Client eine Verbindung zur Tivoli Inventory-Datenbank auf dem DB2-Server herstellen kann, indem Sie den folgenden Befehl eingeben:

   ```
   connect to Aliasname Benutzer Name using Kennwort
   ```

   Dabei gilt:

   Aliasname
   : Steht für den Aliasnamen der Datenbank im Systemdatenbankverzeichnis.

   Name
   : Steht für die ID des Benutzers, der Eigner der Server-Datenbank ist.

Kennwort

Steht für das Kennwort des über *Name* angegebenen Benutzers.

3. Prüfen Sie während der DB2-Sitzung, ob das Konfigurations-Repository installiert wurde, indem Sie den folgenden Befehl eingeben:

   ```
   select * from INVENTORYDATA
   ```

   Als Ergebnis sollten Nullzeilen gefunden werden. Lautet das Ergebnis jedoch, dass die Ansicht INVENTORYDATA unbekannt ist, wurde das Konfigurations-Repository nicht ordnungsgemäß installiert. Überprüfen Sie in diesem Fall die Protokolldateien auf weitere Informationen.

4. Geben Sie zum Beenden der DB2-Sitzung den folgenden Befehl ein:

   ```
   quit
   ```

Weitere Vorgehensweise

Wenn Sie das DB2-Konfigurations-Repository erstellt haben, installieren Sie die Softwaresignaturen und die Tivoli Inventory- und Subskriptionsabfragen. Weitere Informationen können Sie den Abschnitten „Softwaresignaturen installieren” auf Seite 85 und „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87 entnehmen.

Informix-Datenbank als Konfigurations-Repository erstellen

In diesem Abschnitt werden die einzelnen Schritte erläutert, die bei der Erstellung eines Konfigurations-Repositorys auf der Basis einer Informix-Datenbank erforderlich sind. Bevor Sie eine Informix-Datenbank als Konfigurations-Repository erstellen, prüfen Sie folgende Vorbedingungen:

- RIM for Informix verwendet die Befehlszeilenschnittstelle (CLI) von Informix. Hierbei handelt es sich um die Informix-spezifische Adaption von ODBC. In diesem Release von Tivoli Inventory wird die CLI Version 2.5. verwendet

- Der Dämon für RIM for Informix muss als Benutzer 'informix' ausgeführt werden, während die meisten anderen Tivoli-Dämonserver als Benutzer 'nobody' ausgeführt werden (bzw. 'tmersrvd' unter HP-UX und Windows NT). Auf dem RIM-Host muss ein Benutzer mit dem Namen 'informix' vorhanden sein.
- Die CLI von Informix muss auf dem RIM-Host installiert sein und die Dateien '.odbc.ini' und '.odbcinst.ini' müssen in den folgenden Verzeichnissen installiert sein:
  - Im Root-Verzeichnis
  - Im Basisverzeichnis des Informix-Benutzers
  - Im Verzeichnis $INFORMIXDIR
- Es wird empfohlen, die Datenbank im ANSI-Modus auszuführen.
- Befolgen Sie die mit Informix zur Verfügung gestellten Installationsanweisungen. Informix muss unter UNIX in der folgenden Reihenfolge installiert werden:
  1. Eingebettete SQL-Produkte (ESQL = embedded SQL) von Informix
  2. Informix CONNECT
  3. Informix CLI 2.5
  4. Informix-Server

Weitere Informationen können Sie der Dokumentation zu Informix entnehmen.

Informix-Benutzer und Tivoli Inventory-Datenbank erstellen

Weitere Informationen zum Erstellen der Datenbank und des Datenbank-Benutzers können Sie der Dokumentation zu Informix entnehmen. Wenn Sie die Größe der Datenbank festlegen, müssen Sie jeder Maschine, die Sie durchsuchen möchten, 1 Megabyte (MB) zuordnen.

Mit dem Befehl **wsetrimpw** können Sie das Standardkennwort 'tivoli' durch das Kennwort des Exemplareigners ersetzen. Bei dem

Exemplareigner handelt es sich um den Benutzer, den Sie bei Erstellung der Informix-Datenbank und des Informix-Datenbankexemplars angegeben haben. Der Exemplareigner wurde auch als Benutzername im Feld 'User ID' im Dialog 'Installationsoptionen' angegeben.

Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Konfigurations-Repository-Schema installieren

Nachdem Sie Tivoli Inventory installiert haben, erstellen Sie ein Tivoli Inventory-Konfigurations-Repository auf dem Informix-Datenbanksystem, indem Sie folgende Schritte ausführen:

1. Stellen Sie sicher, dass Sie mindestens 20.000 Sperren in der Datei onconfig gesetzt haben, bevor Sie das Skript zur Installation des Schemas ausführen. Hierbei handelt es sich auch um die Mindestanzahl von Sperren, die für Tivoli Inventory erforderlich ist.
2. Kopieren Sie das Skript 'inv_infx_schema.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS in ein temporäres Verzeichnis, in dem Informix auf dem Informix-Server ausgeführt werden kann.
3. Führen Sie das Skript 'inv_infx_schema.sql' in dem Verzeichnis aus, in dem es
   sich jetzt befindet. Geben Sie den folgenden Befehl ein:

   ```
   dbaccess Datenbank@$INFORMIXSERVER inv_infx_schema.sql >> \
   inv_infx_schema.log 2>&1
   ```

   Dabei steht *Datenbank* für den Namen des Konfigurations-Repositorys von Tivoli Inventory.
4. Wenn Sie Protokolltabellen verwenden, wiederholen Sie die Schritte 1 bis 3 mit dem Skript 'h_inv_infx_schema.sql'.

Konfiguration prüfen

Folgende Schritte sind erforderlich, um die Verbindung zwischen dem RIM-Host und dem RDBMS-Server zu überprüfen und um festzustellen, ob das Konfigurations-Repository erstellt wurde:

1. Starten Sie auf dem RIM-Host eine interaktive SQL-Sitzung, indem Sie den folgenden Befehl eingeben:

   ```
   dbaccess Datenbank@$INFORMIXSERVER
   ```

   Dabei steht *Datenbankname* für den Namen der Datenbank und *INFORMIXSERVER* für den Namen des Informix-Servers.

2. Wählen Sie in der Informix-Sitzung 'Abfrage-Sprache' aus.
3. Wählen Sie 'Neu' aus.
4. Geben Sie folgenden Befehl ein:

   ```
   select * from INVENTORYDATA
   ```

5. Wählen Sie 'Ausführen' aus, um zu überprüfen, ob das Konfigurations-Repository installiert wurde.

   Als Ergebnis sollten Nullzeilen gefunden werden. Lautet das Ergebnis jedoch, dass die Ansicht INVENTORYDATA unbekannt ist, wurde das Konfigurations-Repository nicht ordnungsgemäß installiert. Überprüfen Sie in diesem Fall die Protokolldateien auf weitere Informationen.

6. Wählen Sie 'Beenden' aus, um die Informix-Sitzung zu beenden.

Weitere Vorgehensweise

Nachdem Sie das Informix-Konfigurations-Repository erstellt haben, installieren Sie die Softwaresignaturen und die Tivoli Inventory- und Subskriptionsabfragen. Weitere Informationen können sie den Abschnitten „Softwaresignaturen installieren" auf Seite 85 und „Tivoli Inventory- und Subskriptionsabfragen installieren" auf Seite 87 entnehmen.

MS SQL-Server als Konfigurations-Repository erstellen

In diesem Abschnitt werden die einzelnen Schritte erläutert, die bei der Erstellung eines Konfigurations-Repositorys auf der Basis einer MS SQL-Datenbank erforderlich sind. Stellen Sie vor Erstellung des Konfigurations-Repositorys zunächst sicher, dass der Pfad für 'isql' in der Umgebung gesetzt ist, in der die Skripts ausgeführt werden.

Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen

Sie müssen das Skript 'inv_ms_sql_admin.sql' als Schablone verwenden, wenn Sie eine Datenbank, den Benutzer 'invtiv' und das Kennwort 'tivoli' auf dem MS SQL-Server-Datenbanksystem erstellen.

Standardmäßig wird von dem Skript eine Datenbank mit 20 MB erstellt, in der Informationen zu ca. 40 PCs gespeichert werden können. Für jedes System, zu dem Informationen in der Datenbank gespeichert werden, sollten 500 KB zur Verfügung gestellt werden. Für ein Konfigurations-Repository mit Informationen zu 1000 Knoten ist daher ein Platzbedarf von 500 MB erforderlich. Standardmäßig wird vom Skript 'inv_ms_sql_admin.sql' für die Erstellung des Konfigurations-Repositorys die Haupt*einheit* verwendet. Bei MS SQL-Servern bezieht sich der Begriff Einheit auf den Bereich auf einem Server, in dem Benutzer eine Datenbank einrichten können. Die Haupteinheit muss für die Hauptdatenbank und die Systemfunktionen reserviert sein. Daher müssen Sie das Skript vor dessen Ausführung wie folgt ändern:

1. Erstellen Sie eine neue Einheit. Führen Sie dazu die folgenden Schritte aus:
   a. Legen Sie die Größe der neuen Einheit für das Tivoli Inventory-Konfigurations-Repository fest. Sie sollte mindestens so groß sein wie die Datenbank, die Sie bei Ausführung des Skripts 'inv_ms_sql_admin.sql' erstellen.
   b. Erstellen Sie die neue Einheit. Weitere Informationen zur Erstellung neuer Einheiten können Sie der Dokumentation zum MS SQL-Server entnehmen.
2. Kopieren Sie das Skript 'inv_ms_sql_admin.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis auf einem System mit der MS SQL Client-Software.
3. Bearbeiten Sie das Skript, um die unter Schritt 1 erstellte Einheit verwenden zu können.

   Ändern Sie dazu die Zeile:

```
create database inv_db on master = 20
```

in:

```
create database inv_db on Neue_Einheit = Größe
```

Dabei gilt:

Neue_Einheit
: Steht für den Namen der unter Schritt 1 auf Seite 73 erstellten Einheit.

Größe
: Steht für die Größe der Datenbank, die bei Ausführung des Skripts erstellt wird. Die Größe der Datenbank beträgt maximal die Größe von *Neue_Einheit*.

Nach Angabe der gewünschten Einheit und deren Größe im Skript sind zur Erstellung des Konfigurations-Repositorys folgende Schritte erforderlich:

1. Führen Sie unter dem Benutzernamen 'sa' das Skript 'inv_ms_sql_admin.sql' in dem Verzeichnis aus, in dem es sich jetzt befindet. Geben Sie den folgenden Befehl ein:

   ```
   isql -U sa [-P Kennwort] -i inv_ms_sql_admin.sql \
   -o inv_ms_sql_admin.log
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des RDBMS-Benutzers 'sa'.

   Daraufhin wird von dem Skript der Benutzer 'invtiv' (mit dem Kennwort 'tivoli') und die Datenbank 'inv_db' auf dem MS SQL-Server-Datenbanksystem erstellt. Die Ausgabe wird von diesem Skript in der Datei 'inv_ms_sql_admin.log' aufgezeichnet. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht.

2. Aus Sicherheitsgründen wird empfohlen, das Kennwort für den Benutzer 'invtiv' zu ändern. Geben Sie dazu den entsprechenden isql-Befehl ein, und melden Sie anschließend unter Verwendung des Befehls **wsetrimpw** die Kennwortänderung an Tivoli Management Framework und Tivoli Inventory.

Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Konfigurations-Repository-Schema installieren

Folgende Schritte sind zur Ausführung des Skripts 'inv_ms_sql-_schema.sql' erforderlich, mit dem das Schema des Konfigurations-Repositorys installiert wird. Dieses Schema definiert die Tabellen und Ansichten im Konfigurations-Repository. Sie müssen das Skript als RDBMS-Benutzer 'invtiv' ausführen. Folgende Schritte sind erforderlich:

1. Kopieren Sie das Skript 'inv_ms_sql_schema.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis, von dem aus 'isql' auf einem System mit der MS SQL Client-Software ausgeführt werden kann.

2. Führen Sie unter dem Benutzernamen 'invtiv' das Skript 'inv_ms_sql_schema.sql' in dem Verzeichnis aus, in dem es sich jetzt befindet. Geben Sie den folgenden Befehl ein:

   ```
   isql -U invtiv [-P Kennwort] \
   -i inv_ms_sql_schema.sql \
   -o inv_ms_sql_schema.log
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des MS SQL Server-Benutzers 'invtiv'.

   Daraufhin werden von diesem Skript die Tabellen und Ansichten des Konfigurations-Repositorys installiert und die Ausgabe in der Datei 'inv_ms_sql_schema.log' aufgezeichnet. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht.

3. Wenn Sie Protokolltabellen verwenden, wiederholen Sie die Schritte 1 bis 2 mit dem Skript 'h_inv_ms_sql_schema.sql' und der zugehörigen Protokolldatei 'h_inv_ms_sql_schema.log'.

Konfiguration prüfen

Folgende Schritte sind erforderlich, um die Verbindung zwischen dem RIM-Host und dem RDBMS-Server zu überprüfen und um festzustellen, ob das Konfigurations-Repository erstellt wurde:

1. Prüfen Sie auf dem RIM-Host, ob es sich bei 'invtiv' um einen zulässigen Benutzer des MS SQL-Servers handelt und ob eine Verbindung zwischen dem Client und dem Verwaltungssystem für relationale Datenbanken hergestellt werden kann; geben Sie dazu folgenden Befehl ein:

   ```
   isql -U invtiv -P Kennwort
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des MS SQL Server-Benutzers 'invtiv'.

2. Prüfen Sie während der isql-Sitzung, ob das Konfigurations-Repository installiert wurden, indem Sie den folgenden Befehl eingeben:

   ```
   select * from INVENTORYDATA
   go
   ```

   Als Ergebnis sollten Nullzeilen gefunden werden. Lautet das Ergebnis jedoch, dass die Ansicht INVENTORYDATA unbekannt ist, wurde das Konfigurations-Repository nicht ordnungsgemäß installiert. Überprüfen Sie in diesem Fall die Protokolldateien auf weitere Informationen.

3. Geben Sie zum Beenden der isql-Sitzung den folgenden Befehl ein:

   ```
   exit
   ```

Weitere Vorgehensweise

Nachdem Sie das Konfigurations-Repository des MS SQL-Servers erstellt haben, installieren Sie die Softwaresignaturen und die Tivoli Inventory- und Subskriptionsabfragen. Weitere Informationen können Sie den Abschnitten „Softwaresignaturen installieren" auf Seite 85 und „Tivoli Inventory- und Subskriptionsabfragen installieren" auf Seite 87 entnehmen.

Oracle-Datenbank als Konfigurations-Repository erstellen

In diesem Abschnitt werden die einzelnen Schritte erläutert, die bei der Erstellung eines Konfigurations-Repositorys auf der Basis einer Oracle-Datenbank erforderlich sind. Vor Erstellung des Konfigurations-Repositorys ist jedoch zunächst Folgendes zu prüfen bzw. auszuführen:

- Geben Sie für die Umgebungsvariable ORACLE_HOME den Pfad für das Verzeichnis an, in dem die Software für den Oracle-Client bzw. -Server installiert ist.
- Wenn sich der RIM-Host nicht auf derselben Maschine wie der RDBMS-Server befindet, müssen Sie die Umgebungsvariable TWO_TASK setzen. Hierbei handelt es sich um eine Kennung, unter der vom Client Informationen aus der Datei 'tnsnames.ora' abgerufen werden.
- Stellen Sie sicher, dass der Pfad für sqlplus in der Umgebung angegeben wird, in der die Skripts ausgeführt werden.

Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen

Folgende Schritte sind zur Ausführung des Skripts 'inv_ora_admin.sql' erforderlich, das den für das Konfigurations-Repository erforderlichen Tabellenbereich sowie den temporären Speicherbereich für den Benutzer 'invtiv' mit dem Kennwort 'tivoli' auf dem Oracle-Datenbanksystem erstellt. Von diesem Skript wird ein Tabellenbereich mit 50 MB erstellt. Für jedes System, zu dem Informationen in der Datenbank gespeichert werden, sollten 500 KB zur Verfügung gestellt werden. Für ein Konfigurations-Repository mit Informationen zu 1000 Knoten ist daher ein Platzbedarf von 500 MB erforderlich. Zum Erstellen größerer Tabellenbereiche muss das Skript 'inv_ora_admin.sql' entsprechend angepasst werden.

Sie können folgende Änderungen in Skript 'inv_ora_admin.sql' vornehmen:

- Größe des Tabellenbereichs; ersetzen Sie dazu die Zeile:

```
datafile 'inv_data_1' size 50M
```

durch:

```
datafile 'inv_data_1' size Größe_des_Tabellenbereichs
```

Dabei steht *Größe_des_Tabellenbereichs* für die gewünschte Größe des Tabellenbereichs.

- Größe des temporären Tabellenbereichs; ersetzen Sie dazu die Zeile:

  ```
  datafile 'inv_temp_1' size 10M
  ```

  durch:

  ```
  datafile 'inv_temp_1' size Größe_des_temp_Tabellenbereichs
  ```

  Dabei steht *Größe_des_temp_Tabellenbereichs* für die gewünschte Größe des temporären Tabellenbereichs.

Folgende Schritte sind zur Erstellung der für das Konfigurations-Repository von Tivoli Inventory erforderlichen Tabellenbereiche erforderlich:

1. Kopieren Sie das Skript ’inv_ora_admin.sql’ aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis auf einem System mit der Oracle Client-Software.
2. Starten Sie eine SQL*Plus-Sitzung von dem Verzeichnis aus, in dem sich das Skript jetzt befindet, und melden Sie sich am Verwaltungssystem für relationale Datenbanken als Benutzer ’system’ an. Geben Sie den folgenden Befehl ein:

   ```
   sqlplus system/Kennwort
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des RDBMS-Benutzers ’system’.
3. Geben Sie folgenden Befehl ein, wenn die Protokolldaten in einer Datei gespeichert werden sollen:

   ```
   spool inv_ora_admin.log
   ```
4. Führen Sie in der SQL*Plus-Sitzung das Skript ’inv_ora_admin.sql’ aus. Geben Sie den folgenden Befehl ein:

   ```
   @inv_ora_admin.sql
   ```

Daraufhin werden vom Skript der Benutzer 'invtiv'(mit dem Kennwort 'tivoli') und die für das Konfigurations-Repository erforderlichen Tabellenbereiche erstellt. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht.

5. Geben Sie zum Beenden der SQL*Plus-Sitzung nach Ausführung des Skripts den folgenden Befehl ein:

   ```
   quit
   ```

6. Aus Sicherheitsgründen wird empfohlen, das Kennwort für den Benutzer 'invtiv' zu ändern. Geben Sie dazu den entsprechenden SQL*Plus-Befehl ein, und melden Sie anschließend unter Verwendung des Befehls **wsetrimpw** die Kennwortänderung an Tivoli Management Framework und Tivoli Inventory.

   Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Konfigurations-Repository-Schema installieren

Folgende Schritte sind zur Ausführung des Skripts 'inv_ora_schema.sql' erforderlich, mit dem das Schema des Konfigurations-Repositorys installiert wird. Dieses Schema definiert die Tabellen und Ansichten im Konfigurations-Repository. Sie müssen das Skript als RDBMS-Benutzer 'invtiv' ausführen. Folgende Schritte sind erforderlich:

1. Kopieren Sie das Skript 'inv_ora_schema.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis auf einem System mit der Oracle Client-Software.

2. Starten Sie eine SQL*Plus-Sitzung von dem Verzeichnis aus, in dem sich das Skript jetzt befindet, und melden Sie sich am RDBMS-Server als RDBMS-Benutzer 'invtiv' an. Geben Sie den folgenden Befehl ein:

   ```
   sqlplus invtiv/Kennwort
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des Oracle-Benutzers 'invtiv'.

3. Geben Sie folgenden Befehl ein, wenn die Protokolldaten in einer Datei gespeichert werden sollen:

   ```
   spool inv_ora_schema.log
   ```

4. Führen Sie in der SQL*Plus-Sitzung das Skript 'inv_ora_schema.sql' aus. Geben Sie den folgenden Befehl ein:

   ```
   @inv_ora_schema.sql
   ```

   Daraufhin werden von diese Skript die Tabellen und Ansichten des Konfigurations-Repositorys installiert. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht.

5. Geben Sie zum Beenden der SQL*Plus-Sitzung nach Ausführung des Skripts den folgenden Befehl ein:

   ```
   quit
   ```

6. Wenn Sie Protokolltabellen verwenden, wiederholen Sie die Schritte 1 bis 5 mit dem Skript 'h_inv_ora_schema.sql' und der zugehörigen Protokolldatei 'h_inv_ora_schema.log'.

Konfiguration prüfen

Folgende Schritte sind erforderlich, um die Verbindung zwischen dem RIM-Host und dem RDBMS-Server zu überprüfen und um festzustellen, ob das Konfigurations-Repository erstellt wurde:

1. Prüfen Sie auf dem RIM-Host, ob es sich bei 'invtiv' um einen zulässigen Oracle-Benutzer handelt und ob eine Verbindung zwischen dem RIM-Host und der Tivoli Inventory-Datenbank auf dem Oracle-Server hergestellt werden kann; geben Sie dazu den folgenden Befehl ein:

   ```
   sqlplus invtiv/Kennwort
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des Oracle-Benutzers 'invtiv'.

2. Prüfen Sie in der SQL*Plus-Sitzung, ob das Konfigurations-Repository auf dem Oracle-Datenbanksystem installiert wurde, indem Sie den folgenden Befehl eingeben:

   ```
   select * from INVENTORYDATA;
   ```

Als Ergebnis sollten Nullzeilen gefunden werden. Lautet das Ergebnis jedoch, dass die Ansicht INVENTORYDATA unbekannt ist, wurde das Konfigurations-Repository nicht ordnungsgemäß installiert. Überprüfen Sie in diesem Fall die Protokolldateien auf weitere Informationen.

3. Geben Sie zum Beenden der SQL*Plus-Sitzung den folgenden Befehl ein:

   ```
   quit
   ```

Weitere Vorgehensweise

Nachdem Sie das Oracle-Konfigurations-Repository erstellt haben, installieren Sie die Softwaresignaturen und die Tivoli Inventory- und Subskriptionsabfragen. Weitere Informationen können Sie den Abschnitten „Softwaresignaturen installieren” auf Seite 85 und „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87 entnehmen.

Sybase-Datenbank als Konfigurations-Repository erstellen

In diesem Abschnitt werden die einzelnen Schritte erläutert, die bei der Erstellung eines Konfigurations-Repositorys auf der Basis einer Sybase-Datenbank erforderlich sind. Vor Erstellung des Konfigurations-Repositorys ist jedoch zunächst Folgendes zu prüfen bzw. auszuführen:

- Geben Sie für die Umgebungsvariable SYBASE den Pfad für das Verzeichnis an, in dem die Software für den Sybase-RDBMS-Client bzw. -Server installiert ist.
- Geben Sie für die Umgebungsvariable DSQUERY die Kennung an, unter der der Client Verbindungsinformationen aus der Datei ’interfaces’ abruft.
- Stellen Sie sicher, dass der Pfad für ’isql’ in der Umgebung angegeben wird, in der die Skripts ausgeführt werden.

Benutzer 'invtiv' und Tivoli Inventory-Datenbank erstellen

Sie müssen für die Erstellung einer Datenbank und des Benutzers 'invtiv' mit dem Kennwort 'tivoli' auf dem Sybase-Datenbanksystem das Skript 'inv_syb_admin.sql' als Schablone verwenden.

Standardmäßig wird von dem Skript eine Datenbank mit 20 MB erstellt, in der Informationen zu ca. 40 PCs gespeichert werden können. Für jedes System, zu dem Informationen in der Datenbank gespeichert werden, sollten 500 KB zur Verfügung gestellt werden. Für ein Konfigurations-Repository mit Informationen zu 1000 Knoten ist daher ein Platzbedarf von 500 MB erforderlich.

Standardmäßig wird von dem Skript 'inv_syb_admin.sql' die Haupteinheit für die Erstellung des Konfigurations-Repositorys verwendet. Bei Sybase bezieht sich der Begriff *Einheit* auf den Bereich auf einem Server, in dem Benutzer eine Datenbank einrichten können. Die Haupteinheit muss für die Hauptdatenbank und die Systemfunktionen reserviert sein. Daher müssen Sie das Skript vor seiner Ausführung wie folgt ändern:

1. Erstellen Sie eine neue Einheit. Führen Sie dazu die folgenden Schritte aus:

   a. Legen Sie die Größe der neuen Einheit für das Sybase-Konfigurations-Repository fest. Es sollte mindestens so groß wie die Datenbank sein, die Sie bei Ausführung des Skripts unter Schritt 4 erstellen.

   b. Erstellen Sie die neue Einheit. Weitere Informationen zur Erstellung neuer Einheiten können Sie der Dokumentation zu Sybase entnehmen.

2. Kopieren Sie das Skript 'inv_syb_admin.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis auf einem System mit der Sybase Client-Software.

3. Ändern Sie das Skript, indem Sie die unter Schritt 1 erstellte Einheit angeben. Ersetzen Sie dazu die Zeile:

   ```
   create database inv_db on master = 20
   ```

durch:

```
create database inv_db on Neue_Einheit = Größe
```

Dabei gilt:

Neue_Einheit
: Steht für den Namen der unter Schritt 1 erstellten Einheit.

Größe
: Steht für die Größe der Datenbank, die bei Ausführung des Skripts erstellt wird. Die Größe der Datenbank beträgt maximal die Größe von *Neue_Einheit*.

Nach Angabe der gewünschten Einheit und deren Größe im Skript sind zur Erstellung des Konfigurations-Repositorys folgende Schritte erforderlich:

1. Führen Sie unter dem Benutzernamen ’sa’ das Skript ’inv_syb_admin.sql’ in dem Verzeichnis aus, in dem es sich jetzt befindet. Geben Sie den folgenden Befehl ein:

   ```
   isql -U sa -P Kennwort -i inv_syb_admin.sql \
   -o inv_syb_admin.log
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des RDBMS-Benutzers ’sa’.

   Daraufhin werden von dem Skript der Benutzer ’invtiv’ (mit dem Kennwort ’tivoli’) sowie die Datenbank ’inv_db’ auf dem Sybase-Datenbanksystem erstellt, und die Ausgabe wird in der Datei ’inv_syb_admin.log’ aufgezeichnet.

2. Aus Sicherheitsgründen wird empfohlen, das Kennwort für den Benutzer ’invtiv’ zu ändern. Geben Sie dazu den entsprechenden isql-Befehl ein, und verwenden Sie anschließend den Befehl **wsetrimpw**, um die Kennwortänderung an Tivoli Management Framework und Tivoli Inventory zu melden.

   Weitere Informationen zum Befehl **wsetrimpw** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Konfigurations-Repository-Schema installieren

Folgende Schritte sind zur Ausführung des Skripts 'inv_syb_schema.sql' erforderlich, mit dem das Schema des Konfigurations-Repositorys installiert wird. Dieses Schema definiert die Tabellen und Ansichten im Konfigurations-Repository. Sie müssen das Skript als RDBMS-Benutzer 'invtiv' ausführen. Folgende Schritte sind erforderlich:

1. Kopieren Sie das Skript 'inv_syb_schema.sql' aus dem Verzeichnis $BINDIR/../generic/inv/SCRIPTS/RDBMS auf dem TMR-Server in ein temporäres Verzeichnis, in dem 'isql' auf einem System mit der Sybase Client-Software ausgeführt werden kann.
2. Führen Sie unter dem Benutzernamen 'invtiv' das Skript 'inv_syb_schema.sql' in dem Verzeichnis aus, in dem es sich jetzt befindet. Geben Sie folgenden Befehl ein:

   ```
   isql -U invtiv -P Kennwort -i inv_syb_schema.sql \
   -o inv_syb_schema.log
   ```

   Dabei steht *Kennwort* für das RDBMS-Kennwort des Sybase-Benutzers 'invtiv'.

   Daraufhin werden vom Skript die Tabellen und Ansichten für das Konfigurations-Repository installiert und die Ausgabe in der Datei 'inv_syb_schema.log' aufgezeichnet.
3. Wenn Sie Protokolltabellen verwenden, wiederholen Sie die Schritte 1 bis 2 mit dem Skript 'h_inv_syb_schema.sql' und der zugehörigen Protokolldatei 'h_inv_syb_schema.log'.

Konfiguration prüfen

Folgende Schritte sind erforderlich, um die Verbindung zwischen dem RIM-Host und dem RDBMS-Server zu überprüfen und um festzustellen, ob das Konfigurations-Repository erstellt wurde:

1. Prüfen Sie auf dem RIM-Host, ob es sich bei 'invtiv' um einen zulässigen Sybase-Benutzer handelt und ob eine Verbindung zwischen dem RIM-Host und dem Sybase-Datenbanksystem hergestellt werden kann; geben Sie dazu den folgenden Befehl ein:

   ```
   isql -U invtiv -P Kennwort
   ```

Dabei steht *Kennwort* für das RDBMS-Kennwort des Sybase-Benutzers 'invtiv'.

2. Prüfen Sie in der isql-Sitzung, ob das Konfigurations-Repository auf dem Sybase-Datenbanksystem installiert wurde, indem Sie den folgenden Befehl eingeben:

   ```
   select * from INVENTORYDATA
   go
   ```

   Als Ergebnis sollten Nullzeilen gefunden werden. Lautet das Ergebnis jedoch, dass die Ansicht INVENTORYDATA unbekannt ist, wurde das Konfigurations-Repository nicht ordnungsgemäß installiert. Überprüfen Sie in diesem Fall die Protokolldateien auf weitere Informationen.

3. Geben Sie zum Beenden der isql-Sitzung den folgenden Befehl ein:

   ```
   quit
   ```

Weitere Vorgehensweise

Nachdem Sie das Sybase-Konfigurations-Repository erstellt haben, installieren Sie die Softwaresignaturen und die Tivoli Inventory- und Subskriptionsabfragen. Weitere Informationen können Sie den Abschnitten „Softwaresignaturen installieren" und „Tivoli Inventory- und Subskriptionsabfragen installieren" auf Seite 87 entnehmen.

Softwaresignaturen installieren

Bei der Installation von Tivoli Inventory muss außerdem eine Gruppe von Softwaresignaturen im Konfigurations-Repository installiert werden. Bei Softwaresignaturen handelt es sich um Daten zur Identifizierung von bestimmten Softwarepaketen wie beispielsweise den Namen und die Größe der ausführbaren Datei für das jeweilige Softwarepaket. Anhand von Softwaresignaturen kann von Tivoli Inventory ermittelt werden, welche Softwarepakete sich auf den Maschinen befinden, auf denen Sie Suchläufe ausführen. Wenn Sie einen Softwaresignaturen-Suchlauf auf einem Endpunkt ausführen, werden die Softwaresignaturen von Tivoli Inventory an den Endpunkt verteilt; anschließend werden alle Dateien auf dem Endpunkt mit der Liste der Softwaresignaturen verglichen. Wenn eine Datei

einer Softwaresignatur entspricht, werden die Daten der Softwaresignatur für diese Datei an das Konfigurations-Repository übermittelt.

Die Softwaresignaturen, die standardmäßig in Tivoli Inventory zur Verfügung gestellt werden, befinden sich in der Datei SWSIGS.INI, die bei der Installation von Tivoli Inventory auf dem TMR-Server und den verwalteten Knoten installiert wird. Sie müssen die Softwaresignaturen mit dem Befehl **winvsig** aus der Datei 'SWSIGS.INI' im Konfigurations-Repository installieren.

In der folgenden Tabelle sind die für die Installation oder Aktualisierung der Tivoli Inventory-Softwaresignaturen erforderlichen Berechtigungsklassen aufgeführt:

Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Tivoli Inventory-Softwaresignaturen installieren	Richtlinienbereich	**RIM_update** oder **super**

Die Installation der Tivoli Inventory-Softwaresignaturen im Konfigurations-Repository erfolgt mit dem Befehl **winvsig** unter Angabe des Arguments **–f**; damit werden die in der Datei 'SWSIGS-.INI' angegebenen Softwaresignaturen hinzugefügt. Geben Sie den folgenden Befehl ein:

```
winvsig -a -f $BINDIR/../generic/inv/SIGNATURES/SWSIGS.INI
```

Dabei gilt:

–a Gibt an, dass dem Konfigurations-Repository Softwaresignaturen hinzugefügt werden sollen.

–f Gibt an, dass sich die Softwaresignaturen, die hinzugefügt werden sollen, in einer Datei befinden.

$BINDIR/**../generic/...**
Gibt das Verzeichnis an, in dem sich die Datei **SWSIGS.INI** befindet.

SWSIGS.INI

Gibt den Namen der Datei mit den Softwaresignaturen an, die im Konfigurations-Repository installiert werden sollen. Die Datei SWSIGS.INI wird zur Installation aller Softwaresignaturen verwendet.

Weitere Informationen zu Softwaresignaturen können Sie dem Kapitel „Benutzerspezifische Daten mit Tivoli Inventory erfassen" auf Seite 121 entnehmen. Weitere Informationen zum Befehl **winvsig** finden Sie im Anhang „Befehle" auf Seite 239.

Tivoli Inventory- und Subskriptionsabfragen installieren

Die bei einem Hardware- bzw. Softwaresuchlauf mit Tivoli Inventory ermittelten Daten können Sie mit der Abfragefunktion von Tivoli Management Framework abrufen. Tivoli Inventory stellt vier Gruppen von vordefinierten Abfragen zur Verfügung: die Inventarabfragen, die Subskriptionsabfragen, die Abfragen zum Inventarprotokoll und die Abfragen zum Subskriptionsprotokoll. Diese Abfragen werden bei der Ausführung der Skripts 'inventory_query.sh', 'subscription_query.sh', 'h_inventory_query.sh' und 'h_subscription_query.sh' installiert, die alle in Tivoli Inventory zur Verfügung stehen. Von jedem Skript wird eine Abfragebibliothek mit einer Gruppe von Abfragen erstellt, mit denen Sie auf die von Tivoli Inventory im Konfigurations-Repository gespeicherten Daten zugreifen können.

- Von dem Skript 'inventory_query.sh' wird die Abfragebibliothek 'INVENTORY_QUERY' erstellt; diese enthält Abfragen, mit denen allgemeine Informationen zu den Maschinen in Ihrer TMR abgerufen werden können.
- Von dem Skript 'subscription_query.sh' wird die Abfragebibliothek 'SUBSCRIPTION_QUERY' erstellt; diese enthält Abfragen, mit denen anhand der Ergebnisse verschiedener Suchläufe Listen möglicher Profilsubskribenten abgerufen werden können.

- Von dem Skript 'h_inventory_query.sh' wird die Abfragebibliothek 'H_INVENTORY_QUERY' erstellt; diese enthält Protokollabfragen, mit denen allgemeine Informationen zum Inventarprotokoll zu den Maschinen in Ihrer TMR abgerufen werden können.
- Von dem Skript 'h_subscription_query.sh' wird die Abfragebibliothek 'H_SUBSCRIPTION_QUERY' erstellt; diese enthält Protokollabfragen, mit denen anhand der Ergebnisse verschiedener Suchläufe Listen möglicher Profilsubskribenten abgerufen werden können.

Die Skripts befinden sich im Verzeichnis *$BINDIR*/../generic/inv/SCRIPTS/QUERIES. Sie haben die Möglichkeit, die beiden Bibliotheken bei der Ausführung der Skripts umzubenennen. Führen Sie die Skripts auf dem TMR-Server für die TMR aus, in der diese Abfragen zur Anwendung kommen sollen. Da die Namen von Abfragen und Abfragebibliotheken in einer TMR eindeutig sein müssen, kann nur jeweils eine Gruppe von Tivoli Inventory-Abfragen in einer TMR bzw. mehreren miteinander verbundene TMRs vorliegen.

Eine Beschreibung der Abfragen, die von den Skripts erstellt werden, können Sie dem Anhang „Konfigurations-Repository" auf Seite 365 entnehmen. Informationen zum Erstellen neuer Abfragen für den Abruf von Tivoli Inventory-Daten können Sie dem Kapitel „Inventory-Daten abfragen" auf Seite 209 entnehmen.

In der folgenden Tabelle sind die für die Erstellung von Tivoli Inventory-Abfragen erforderlichen Berechtigungsklassen aufgeführt:

Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Inventory- und Subskriptionsabfragen erstellen	Richtlinienbereich	**admin**, **senior** oder **super**

Folgende Schritte sind zur Ausführung der Skripts 'inventory_query-.sh', 'subscription_query.sh', 'h_inventory_query.sh' und 'h_subscription_query.sh' erforderlich:

1. Wählen Sie den Richtlinienbereich aus, in dem die Abfragebibliotheken erstellt werden sollen.
2. Fügen Sie dem Richtlinienbereich die verwaltete Ressourcenart 'QueryLibrary' hinzu. Weitere Informationen zum Hinzufügen verwalteter Ressourcen in einem Richtlinienbereich können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.
3. Stellen Sie sicher, dass die Tivoli-Umgebung eingerichtet ist. Ist dies nicht der Fall, führen Sie das Skript '/etc/Tivoli/setup_env.sh' aus.
4. Führen Sie die einzelnen Skripts aus, indem Sie jeweils den folgenden Befehl eingeben:

   ```
   $BINDIR/../generic/inv/SCRIPTS/QUERIES/Skript Bereich \
   Abfragebibliothek
   ```

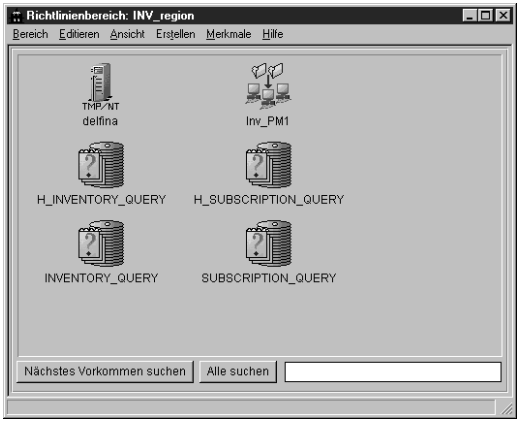
Dabei gilt:

Skript Gibt als Namen des Skripts inventory_query.sh, subscription_query.sh, h_inventory_query.sh oder h_subscription_query.sh an.

Bereich
Steht für den Bereich, in dem die Abfragebibliothek erstellt werden soll.

Abfragebibliothek
(Optional) Steht für den von Ihnen angegebenen Namen, der von dem standardmäßig im Skript angegebenen Namen abweicht.

Nach Ausführung der Skripts enthält das Fenster des Richtlinienbereichs auf der Tivoli-Arbeitsoberfläche vier neue Symbole; siehe das folgende Beispiel:



Weitere Vorgehensweise

Nachdem Sie das Konfigurations-Repository erstellt sowie die Softwaresignaturen und die Abfragen installiert haben, sollten Sie SCS konfigurieren.

4

Mit SCS arbeiten

Nach der Installation von Tivoli Inventory und dem Erstellen des Konfigurations-Repositorys wird empfohlen, Ihre Tivoli-Umgebung für die Verwendung von SCS (Scalable Collection Service) zu konfigurieren. SCS verwendet für die Datenübertragung von Endpunkten an den Inventory Data Handler eine Kollektorhierarchie. Die Daten werden dann über ein oder mehrere RIM-Objekte an das Konfigurations-Repository weitergegeben. Sie sollten die SCS-Komponenten optimieren, um einen effektiven Datenfluss zu gewährleisten und Fehler während der Datenübertragung zu vermeiden. In diesem Kapitel finden Sie eine Beschreibung des gemeinsamen Einsatzes von Tivoli Inventory und SCS sowie Informationen zur Konfiguration und Verwendung von Tivoli Inventory bei einem gemeinsamen Einsatz mit SCS.

Dieses Kapitel enthält folgende Informationen:

- Erfassen von Suchlaufdaten und Speichern der Dateien im Konfigurations-Repository durch Tivoli Inventory und SCS. (Siehe „SCS zusammen mit Tivoli Inventory verwenden” auf Seite 93.)
- Informationen zum Stoppen, Starten, Zurücksetzen und Konfigurieren von Kollektoren. (Siehe „Kollektoren verwenden” auf Seite 95.)

- Informationen zum Konfigurieren des Inventory Data Handler, damit Daten in das RIM-Objekt geschrieben werden, zum Verwalten von Statusinformationen zu den Suchläufen und zum Verschieben des Inventory Data Handler. (Siehe „Den Inventory Data Handler verwenden” auf Seite 99.)
- Die Prozedur zum Verschieben des Callback-Objekts von Inventory. (Siehe „Das Callback-Objekt von Inventory verschieben” auf Seite 103.)
- Informationen zu RIM-Objekten. (Siehe „RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren” auf Seite 105.)
- Die Prozedur zum Steuern der Übertragung von erfassten Suchlaufdaten innerhalb der Datenerfassungshierarchie. (Siehe „Datenerfassung terminieren” auf Seite 109.)
- Informationen zur Steuerung des Datenflusses im Netzwerk durch SCS. (Siehe „Datenfluss im Netzwerk steuern” auf Seite 112.)
- Prozeduren zum Anzeigen von Informationen zu Kollektoren, CTOCs (Collection Tables of Contents), Warteschlangen, zum Inventory Data Handler, zu Suchläufen und Inventory-Profilen sowie Informationen zum Anzeigen von Daten im Konfigurations-Repository. (Siehe „Informationen zu SCS abrufen” auf Seite 114.)
- Erläuterungen zum Konfigurieren von Tivoli Inventory, um Statusinformationen zu Suchläufen zu erfassen und zurückzugeben. (Siehe „Statusinformationen in Tivoli Inventory konfigurieren” auf Seite 117.)
- Informationen zur Verwendung von SCS in mehreren TMRs (Tivoli Management Regions). (Siehe „Hinweise zu miteinander verbundenen TMRs” auf Seite 119.)

SCS zusammen mit Tivoli Inventory verwenden

In der folgenden Tabelle wird das Zusammenspiel zwischen Tivoli Inventory und SCS zur Speicherung von Inventory-Daten im Konfigurations-Repository beschrieben:

Aktion	**Produkt oder Service**
Sie erstellen ein Inventory-Profil.	Tivoli Inventory
Sie verteilen das Profil an Endpunkte.	Tivoli Inventory
Die Profilverteilung ist beendet.	Tivoli Inventory
Die folgenden Vorgänge werden auf einem Endpunkt bei Beendigung einer des Suchlaufs auf dem Endpunkt ausgeführt: 1. Eine MIF-Datei wird erstellt. 2. Die MIF-Datei wird syntaktisch analysiert. 3. Suchlaufdaten werden generiert. 4. Die Suchlaufdaten werden verschlüsselt, komprimiert und in einer Datei gespeichert.	Tivoli Inventory
Wenn der Suchlauf auf einem Endpunkt beendet ist, führt der Endpunkt einen aufwärts gerichteten Aufruf an den Gateway aus und sendet ein CTOC, um eine Datenabfrage anzufordern . **Anmerkung:** Wenn der Endpunkt das CTOC nicht erfolgreich an den Kollektor übermitteln kann, werden die Suchlaufdaten für diesen Endpunkt gesendet, ohne dass SCS verwendet wird.	SCS
Der Gateway sendet das CTOC an den lokalen Kollektor.	SCS

Aktion	Produkt oder Service
Wenn der Kollektor auf dem Gateway das CTOC empfängt, werden folgende Vorgänge ausgeführt: 1. Der Kollektor stellt das CTOC in die Eingabewarteschlange. 2. Der Kollektor führt einen abwärts gerichteten Aufruf an den Endpunkt aus, um die komprimierte und codierte Datendatei abzurufen. 3. Der Kollektor speichert die Datendatei im Kollektordepot. Das CTOC wird von der Eingabewarteschlange in die Ausgabewarteschlange verschoben. 4. Der Kollektor gibt eine Anforderung zur Datenerfassung an den nächsten Kollektor in der Hierarchie aus (an den Kollektor, der sich auf einer höheren Ebene befindet), indem er diesem das CTOC sendet. 5. Der Kollektor auf der höheren Ebene ruft die Daten vom Kollektor der niedrigeren Ebene ab. Wenn die Datenübertragung abgeschlossen ist, wird das CTOC in der Ausgabewarteschlange des Kollektors auf der niedrigeren Ebene gelöscht. 6. Die Schritte 1 bis 5 werden wiederholt bis die Datendatei den Inventory Data Handler erreicht.	SCS
Genauso wie Kollektoren speichert der Inventory Data Handler die CTOCs in einer Eingabewarteschlange und Suchlaufdaten in einem Depot. Der Inventory Data Handler sendet jedoch keine Abrufanforderung an einen anderen Kollektor, sondern dekomprimiert die Suchlaufdaten und übermittelt diese an ein verfügbares RIM-Objekt.	Tivoli Inventory
Von dem RIM-Objekt bzw. den -Objekten werden die Daten an das Konfigurations-Repository gesendet. Der Statuskollektor erfasst für jeden Endpunkt den Endstatus.	Tivoli Inventory

Zum Ändern einer SCS-Hierarchie müssen Sie mit dem Befehl **wrpt** Ihre Repeater-Topologie entsprechend Ihren Bedürfnissen erstellen und anschließend mit dem Befehl **wcollect** unter Verwendung der Option **–r** im modifizierten Teil der Hierarchie die Routing-Topologie der Kollektoren zurücksetzen. (Siehe „Kollektoren verwenden".) Da die MDist 2-Repeater-Hierarchie (MDist 2 = multiplexed distribution) auch von SCS verwendet wird (obwohl die Daten in der entgegengesetzten Richtung fließen), müssen Sie die Hierarchie so konfigurieren, dass diese von beiden Services optimal genutzt werden kann. Weitere Informationen zum Befehl **wrpt** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Kollektoren verwenden

In diesem Abschnitt finden Sie Informationen zu den verschiedenen Tasks, die Sie auf Kollektoren mit dem Befehl **wcollect** ausführen können.

Anmerkung: Bei Verwendung des Befehls **wcollect** müssen Sie den Kollektor angeben, für den Sie die Aktion ausführen möchten. Bei dem Kollektornamen handelt es sich immer um den Namen des verwalteten Knotens oder Gateways, auf dem der Kollektor installiert ist.

Weitere Informationen zum Befehl **wcollect** finden Sie unter „wcollect" auf Seite 247.

Kollektoren stoppen, starten und zurücksetzen

Zur Ausführung bestimmter Tasks unter Verwendung von SCS müssen Sie Kollektoren stoppen und erneut starten. Nach der erneuten Konfiguration von Kollektoren müssen Sie diese beispielsweise stoppen und erneut starten, damit die Änderungen wirksam werden. Auch bei der Erstellung eines neuen Laufzeitverzeichnisses müssen Sie Kollektoren stoppen, damit Sie Daten aus dem alten Verzeichnis in das neue Verzeichnis verschieben können. Auch während der Durchführung von Fehlerbehebungsmaßnahmen müssen Sie Kollektoren möglicherweise stoppen und erneut starten.

SCS bietet zwei Optionen zum Stoppen eines Kollektors unter Verwendung des Befehls **wcollect –h** :

- **Kontrolliert** - Bevor der Kollektor angehalten wird, werden zuerst alle aktiven Datenerfassungsvorgänge beendet.
- **Sofort** - Der Kollektor wird sofort angehalten. Die Verarbeitung aktiver Datenerfassungsvorgänge wird unterbrochen. Wenn Sie den Kollektor erneut starten, werden alle Datenerfassungsvorgänge, die aktiv waren, als der Kollektor angehalten wurde, erneut gestartet, so dass keine Daten verloren gehen.

Wenn Sie einen Kollektor mit der Option **–h** gestoppt haben, müssen Sie die Option **–s** verwenden, um den Kollektor erneut zu starten.

Wenn Sie Änderungen an der Kollektorhierarchie mit dem Befehl **wrpt** vorgenommen haben, müssen Sie die Kollektortopologie für alle von der Änderung betroffenen Kollektoren zurücksetzen. Verwenden Sie die Option **–r** des Befehls **wcollect**, um die lokal zwischengespeicherte Datenerfassungsroute für einen Kollektor zu entfernen. Nach Ausführung dieses Befehls lädt der Kollektor neue Routinginformationen aus dem Datenerfassungsmanager, wenn der Kollektor die nächste Abrufanforderung verarbeitet.

Kollektoren konfigurieren

Nach der Installation von Tivoli Inventory sollten Sie die Kollektoren konfigurieren, um sicherzustellen, dass das Laufzeitverzeichnis und das Depot die richtige Größe haben, die Protokolloptionen Ihren Bedürfnissen entsprechen und die Threads eine optimale Kommunikation zwischen den Kollektoren ermöglichen. Möglicherweise ist es erforderlich, die Kollektoren erneut zu konfigurieren, nachdem Sie Änderungen an der Kollektorhierarchie vorgenommen haben. In den folgenden Abschnitten werden die Konfigurationsoptionen für Kollektoren beschrieben.

Das Laufzeitverzeichnis und das Depot konfigurieren

Das Laufzeitverzeichnis enthält das Depot und die Laufzeitdateien (*.dat- und *.log-Dateien). Standardmäßig befindet sich das Laufzeit-

verzeichnis von SCS im Verzeichnis $DBDIR/mcollect auf allen Kollektoren. Das Depot befindet sich im Laufzeitverzeichnis unter $DBDIR/mcollect/depot.

Sie können das Laufzeitverzeichnis und das Depot verschieben, wenn der Speicherplatz im Verzeichnis $DBDIR begrenzt ist. Beispielsweise können Sie das Depot in ein größeres Dateisystem verschieben, wenn Sie eine große Anzahl von Endpunkten durchsuchen oder wenn das Depot den Plattenspeicherplatz gemeinsam mit oserv benutzt, der ebenfalls Daten in das Verzeichnis $DBDIR schreibt.

Um das Laufzeitverzeichnis und das Depot zu verschieben, müssen Sie zunächst eine neue Adresse für das Laufzeitverzeichnis angeben. Verwenden Sie hierzu die Option **–l** des Befehls **wcollect**. Anschließend müssen Sie das Depot und alle anderen Daten manuell aus dem alten Laufzeitverzeichnis in das neue Verzeichnis verschieben. Das Laufzeitverzeichnis muss sich auf einem stabilen Datenträger befinden, der über genügend freien Speicherplatz verfügt, damit die erfassten Daten permanent gespeichert werden können. Bei diesem Verzeichnis sollte es sich um kein temporäres Verzeichnis handeln. Sie müssen dem Konto 'tmersrvd' bzw. 'nobody' (dem Tivoli-Konto ohne Berechtigungen) in diesem Verzeichnis Schreib-/Lese-Berech-
,br tigungen erteilen.

Anmerkung: Nach Angabe eines neuen Laufzeitverzeichnisses müssen Sie den Kollektor stoppen, bevor Sie das Depot und Daten aus dem alten in das neue Verzeichnis verschieben.

Sie können mit der Option **–z** des Befehls **wcollect** die Größe des Depots ändern. Standardmäßig beträgt die Größe des Depots 40 Megabyte (MB). Sie können das Depot jedoch jederzeit vergrößern. Das Depot kann nur vergrößert und nicht verkleinert werden. Sie können beispielsweise die Größe des Depots zunächst auf 50 MB festlegen und später auf 55 MB heraufsetzen. Es ist jedoch nicht möglich die Größe des Depots auf 45 MB herabzusetzen.

Sie können die Größe der Dateneinheiten, die zwischen Kollektoren übertragen werden, steuern. Diese Dateneinheiten werden

Übertragungsblöcke genannt. Die Größe des Übertragungsblocks wird mit der Option **–c** des Befehls **wcollect** festgelegt. Der Standardwert ist 1 MB.

Wenn ein Kollektor versucht, Suchlaufdaten zu erfassen, das Depot jedoch zu klein ist, um die Daten zu speichern, schlägt die Datenerfassung für den durchsuchten Endpunkt fehl. Weitere Informationen finden Sie unter „Verfahren zur Fehlerbehebung” auf Seite 527.

Die Protokolldatei für Kollektoren konfigurieren

Die Datei ’mcollect.log’ enthält alle Vorgänge, die ein Kollektor während des Datentransports ausführt. Sie können die Anzahl der Daten, die in dieser Datei protokolliert werden, festlegen, indem Sie einen Wert über die Option **–d** des Befehls **wcollect** angeben. Standardmäßig werden nur schwer wiegende Fehler protokolliert. Sie können die maximale Größe dieser Datei über die Option**–g** des Befehls **wcollect** festlegen. Der Standardwert ist 1 MB.

Threads konfigurieren

Threads steuern erfolgreiche Übertragungen zwischen zwei Kollektoren. Kollektoren werden als multithreading-fähige Dämonprozesse ausgeführt. Kollektoren verwenden Threads vorwiegend zur Verwaltung der Vielzahl von Datenströmen, die in die Depots und aus den Depots fließen. Mehrere Konfigurationsoptionen des Befehls **wcollect** stehen in Zusammenhang mit den Threads zwischen Kollektoren. Sie können die maximale Anzahl der Eingabethreads mit der Option **–t** und die maximale Anzahl der Ausgabethreads mit der Option **–o** festlegen. Der Standardwert für beide Threadarten ist 5.

Ein Eingabethread befindet sich im Leerlauf, wenn keine Anforderungen in seiner Warteschlange verarbeitet werden können. Sie können die Zeit, die verstreichen soll, bevor der Eingabethread gestoppt wird, über die Option **–i** festlegen. Die Standardleerlaufzeit beträgt 60 Sekunden. Die Wartezeit eines Threads, wenn das System bzw. Netzwerkressourcen vorübergehend nicht verfügbar sind, können Sie mit der Option **–p** festlegen. Der Standardwert sind 5 Sekunden.

Wiederholungen stehen in Zusammenhang mit Threads. Eine *Wiederholung* ist ein Versuch, der von einem Kollektor unternommen wird,

eine Datenerfassungsanforderung eines anderen Kollektors zu verarbeiten, nachdem der erste Versuch fehlgeschlagen ist. Es gibt viele Gründe dafür, dass Datenübertragungen zwischen Kollektoren bzw. zwischen Kollektoren und Endpunkten fehlschlagen. Wenn beispielsweise ein Thread keine Netzwerkverbindung zu einem anderen Kollektor bzw. Endpunkt aushandeln kann, ist der Thread fehlgeschlagen. Indem Sie Wiederholungen und ein Wiederholungsintervall für Kollektoren-Threads konfigurieren, ist SCS in der Lage, sich nach temporären Fehlerbedingungen wiederherzustellen.

Mit der Option **–m** des Befehls **wcollect** können Sie die maximale Anzahl von Versuchen festlegen, die ein Kollektor unternimmt, um eine Datenerfassungsanforderung von einem Kollektor der untergeordneten Ebene zu verarbeiten, nachdem der erste Versuch fehlgeschlagen ist. Der Standardwert ist 10.

Mit der Option **–e** des Befehls **wcollect** können Sie das Wiederholungsintervall festlegen. Hierbei handelt es sich um die Wartezeit (in Sekunden) des Kollektors, bevor er erneut versucht, eine Verbindung zu einem anderen Kollektor herzustellen. Der Standardwert beträgt 1 Sekunde.

Den Inventory Data Handler verwenden

Der Inventory Data Handler ist der letzte Kollektor in der Kollektorhierarchie von SCS. Der Inventory Data Handler empfängt Daten von Kollektoren und sendet diese Daten an das Konfigurations-Repository.

Der Inventory Data Handler wird automatisch auf dem verwalteten Knoten erstellt, den Sie während der Installation angeben. Erfolgt keine Angabe, wird der Inventory Data Handler auf dem TMR-Server erstellt. Es wird jedoch empfohlen, dass Sie den Inventory Data Handler *nicht* auf demselben verwalteten Knoten wie den TMR-Server installieren.

In diesem Abschnitt wird beschrieben, wie Sie den Inventory Data Handler konfigurieren. Darüber hinaus wird erläutert, wie Sie ihn auf einen anderen verwalteten Knoten verschieben.

Den Inventory Data Handler konfigurieren

Der Inventory Data Handler wird bei der Installation von Tivoli Inventory erstellt. Sie können den Inventory Data Handler mit dem Befehl **wcrtinvdh** auf einen anderen verwalteten Knoten verschieben . Mit dem Befehl **wsetinvdh** können Sie festlegen, wie der Inventory Data Handler Daten in ein RIM-Objekt schreibt. Sie können mit diesem Befehl auch die Statuskollektoroptionen konfigurieren, über die die Erfassung, Speicherung und Verteilung von Benachrichtigungen zum Status der Suchläufe angegeben wird. Mit dem Befehl **wcollect** können Sie Ausgabethreads für den Inventory Data Handler konfigurieren.

In diesem Abschnitt finden Sie eine Beschreibung der Optionen für die Konfiguration des Inventory Data Handler. Weitere Informationen zu den Befehlen **wcrtinvdh**, **wsetinvdh** und **wcollect** können Sie dem Abschnitt „Befehle” auf Seite 239 entnehmen.

Mit der Option **–s** des Befehls **wsetinvdh** können Sie angeben, ob der Inventory Data Handler Statusinformationen speichert. *Statusinformationen* zeigen an, ob der Suchlauf auf allen Endpunkten während einer Profilverteilung beendet ist. Bei Auftreten eines Systemfehlers werden die Statusinformationen automatisch wiederhergestellt, sobald der Inventory Data Handler wieder online ist. Standardmäßig werden Statusinformationen gespeichert.

Über die Option **–d** des Befehls **wsetinvdh** können Sie das Verzeichnis angeben, in dem der Inventory Data Handler Statusinformationen speichert. Standardmäßig werden die Statusinformationen im Verzeichnis $DBDIR/inventory/stat_dir auf dem verwalteten Knoten, auf dem sich der Inventory Data Handler befindet, gespeichert. Die Angabe eines anderen Verzeichnisses ist jedoch möglich.

Über die Option **–n** des Befehls **wsetinvdh** können Sie das Intervall (in Minuten) angeben, in dem Statusinformationen gesendet werden . Wenn Sie beispielswiese den Wert auf 10 gesetzt haben, sendet der Inventory Data Handler alle 10 Minuten eine einzelne Nachricht, ein so genanntes *Bündel*, in dem alle Benachrichtigungen über abgeschlossene Suchläufe enthalten sind. Der Inventory Data Handler sendet diese Bündel an die Benachrichtigungsgruppe von Tivoli

Inventory, an eine Protokolldatei und/oder die TEC-Konsole. An welche Ziele die Statusinformationen gesendet werden, hängt von den Einstellungen der Option **–l** des Befehls **wsetinvglobal** ab. Der Standardwert für die Option **–n** des Befehls **wsetinvdh** ist 10 Minuten.

Über die Option **–q** des Befehls **wsetinvdh** können Sie die maximale Anzahl von Zielen in einem Bündel festlegen . Wenn Sie beispielsweise diesen Wert auf 5 gesetzt haben, bündelt und übermittelt der Inventory Data Handler die Statusinformationen, sobald die Suchläufe auf fünf Zielen abgeschlossen sind. Der Standardwert beträgt 10 Ziele.

Über die Optionen **–n** und **–q** des Befehls **wsetinvdh** und die Option **–n** des Befehls **wsetinvglobal** können Sie Statusinformationen konfigurieren, die in Bündeln übermittelt werden sollen .

Hinweise:

1. Wenn Sie sowohl die Option **–q** *alle_n_Ziele_bündeln* als auch die Option **–n** *alle_n_Minuten_bündeln* auf 0 setzen, werden die Statusinformationen nicht gebündelt, und Sie werden erst benachrichtigt, wenn die Suchläufe auf allen Zielen abgeschlossen sind.
2. Wenn Sie sowohl die Option **–q** *alle_n_Ziele_bündeln* als auch die Option **–n** *alle_n_Minuten_bündeln* auf einen positiven Wert setzen, werden die Statusinformationen gebündelt, sobald die angegebene Anzahl von Zielen oder Minuten erreicht wird.
3. Statusnachrichten werden nur gebündelt, wenn die Option **–n** des Befehls **wsetinvglobal** auf ’BUNDLE’ gesetzt ist.

Über die Option **–o** des Befehls **wcollect** können Sie die Anzahl der Ausgabethreads angeben, die dem Inventory Data Handler zum Schreiben der Daten in RIM-Objekte zur Verfügung stehen. Es wird empfohlen, dass dieser Wert der Summe der RDBMS-Verbindungen entspricht, die für alle RIM-Objekte eingerichtet wurden, die vom Inventory Data Handler verwendet werden. Wenn der Inventory Data Handler beispielsweise zwei RIM-Objekte verwendet, um eine Verbindung zum Verwaltungssystem für relationale Datenbanken herzustellen, und jedes RIM-Objekt über 5 RDBMS-Verbindungen verfügt, müssen Sie die Anzahl der Ausgabethreads für den Inventory Data

Handler auf 10 setzen. Weitere Informationen zum Einrichten von RDBMS-Verbindungen für RIM-Objekte finden Sie unter „RIM-Objekte konfigurieren” auf Seite 107.

Über die Option **–r** des Befehls **wsetinvdh** können Sie angeben, wieviele Versuche der Inventory Data Handler unternimmt, um Daten in ein RIM-Objekt zu schreiben. Wenn die angegebene Anzahl von Wiederholungen erreicht ist, sendet der Inventory Data Handler eine Fehlernachricht. Der Standardwert für diese Option sind 5 Wiederholungen.

Über die Option **–t** des Befehls **wsetinvdh**, die mit der Option **–r** des Befehls **wsetinvdh** verwandt ist, können Sie einen Wert angeben, anhand dessen das Zeitlimit zwischen Wiederholungen berechnet wird. Der Inventory Data Handler berechnet das Zeitlimit unter Verwendung des Algorithmus *Zeitlimit*Wiederholungszähler*. Wenn Sie beispielsweise diesen Wert auf 20 Sekunden setzen, wird der Zeitgeber beim ersten Wiederholungsversuch auf 20 * 1 oder 20 Sekunden gesetzt. Beim zweiten Wiederholungsversuch wird der Zeitgeber auf 20 * 2 bzw. 40 Sekunden gesetzt. Der Standardwert für die Option **–t** des Befehls **wsetinvdh** beträgt 30 Sekunden.

Den Inventory Data Handler verschieben

Wenn Sie den Inventory Data Handler verschieben möchten, müssen Sie ihn von dem verwalteten Knoten, auf dem er sich derzeit befindet, löschen. Erstellen Sie anschließend einen neuen Inventory Data Handler auf einem anderen verwalteten Knoten.

Anmerkung: Stellen Sie vor dem Verschieben des Inventory Data Handler sicher, dass keine Inventory-Suchläufe aktiv sind. Wenn Sie den Inventory Data Handler vor Beendigung der Suchläufe verschieben, gehen alle Daten verloren, die das Konfigurations-Repository noch nicht erreicht haben.

Führen Sie zum Verschieben des Inventory Data Handler folgende Tasks aus:

1. Stellen Sie sicher, dass die SCS-Programmkorrektur und das Tivoli Inventory-Produkt auf dem Knoten, auf dem Sie den neuen Inventory Data Handler erstellen möchten, installiert sind.
2. Löschen Sie den vorhandenen Inventory Data Handler mit dem folgenden Befehl:

   ```
   wdel @InvDataHandler:inv_data_handler
   ```

   Weitere Informationen zum Befehl **wdel** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.
3. Erstellen Sie den neuen Inventory Data Handler mit dem Befehl **wcrtinvdh** und den entsprechenden Optionen. Weitere Informationen finden Sie unter „wcrtinvdh” auf Seite 258.

Das Callback-Objekt von Inventory verschieben

Das Callback-Objekt von Inventory hat die folgenden Funktionen:

- Wenn Inventory-Daten nicht über die Kollektorhierarchie erfasst werden können, sendet MDist 2 die Daten an das Callback-Objekt von Inventory, das die Daten an den Inventory Data Handler sendet. Die Daten werden dann über ein oder mehrere RIM-Objekte an das Konfigurations-Repository gesendet.
- Bei allen Suchläufen sendet MDist 2 eine Statusnachricht an das Callback-Objekt von Inventory, die anzeigt, ob das Inventory-Profil erfolgreich zum Endpunkt gesendet werden konnte. Diese Nachricht zeigt nur an, dass der Endpunkt das Profil verarbeitet hat. Ob die Suchlaufdaten das Konfigurations-Repository erreicht haben, wird nicht angezeigt.

Bei der Installation von Tivoli Inventory wird das Callback-Objekt von Inventory automatisch auf dem angegebenen Knoten erstellt. Wenn Sie das Callback-Objekt von Inventory verschieben möchten, müssen Sie es von dem verwalteten Knoten, auf dem es sich gerade befindet, löschen. Erstellen Sie anschließend eine neues Callback-Objekt auf einem anderen verwalteten Knoten. Sie können das Callback-Objekt von Inventory auf jedem beliebigen Knoten in Ihrer TMR erstellen.

Wenn Sie die das Callback-Objekt von Inventory verschieben möchten, müssen Sie es von dem verwalteten Knoten, auf dem es sich gerade befindet, löschen. Erstellen Sie anschließend eine neues Callback-Objekt auf einem anderen verwalteten Knoten.

Anmerkung: Stellen Sie vor dem Verschieben des Callback-Objekts von Inventory sicher, dass keine Inventory-Suchläufe aktiv sind.

Führen Sie zum Verschieben des Callback-Objekts von Inventory folgende Tasks aus:

1. Stellen Sie sicher, dass die SCS-Programmkorrektur und das Tivoli Inventory-Produkt auf dem Knoten, auf dem Sie das neue Callback-Objekt von Inventory erstellen möchten, installiert sind.
2. Löschen Sie das vorhandene Callback-Objekt von Inventory mit dem folgenden Befehl:

   ```
   wdel @InventoryConfigCB:inv_cb
   ```

   Weitere Informationen zum Befehl **wdel** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.
3. Erstellen Sie das neue Callback-Objekt von Inventory mit dem Befehl **wcrtinvcb** und den entsprechenden Optionen. Weitere Informationen zu diesem Befehl können Sie dem Abschnitt „wcrtinvcb” auf Seite 256 entnehmen.

RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren

Der Inventory Data Handler schreibt mit Hilfe der RIM-Objekte Daten in das Konfigurations-Repository. Bei der Installation von Tivoli Inventory wird das RIM-Objekt 'invdh_1' zu diesem Zweck erstellt. Sie können jedoch weitere RIM-Objekte für den Inventory Data Handler erstellen. In diesem Abschnitt finden Sie Informationen zum Erstellen von zusätzlichen RIM-Objekten.

Sie erstellen zusätzliche RIM-Objekte aus folgenden Gründen:

- Um die Anzahl der Verbindungen zum Verwaltungssystem für relationale Datenbanken zu erhöhen. Maximal sind 16 Verbindungen pro RIM-Objekt zulässig.
- Um die Arbeit, die von RIM-Objekten ausgeführt wird, auf mehrere Systeme zu verteilen. Beispielsweise können Sie RIM-Objekte auf separaten verwalteten Knoten erstellen, so dass die Prozesse, die von den RIM-Objekten ausgeführt werden, auf zwei Systeme verteilt werden.

RIM-Objekte erstellen

RIM-Objekte werden mit dem Befehl **wcrtrim** erstellt. Bei der Erstellung von RIM-Objekten für den Inventory Data Handler müssen Sie unter Verwendung der Option **–a** die Anwendungskennung des RIM-Objekts angeben. Darüber hinaus müssen Sie über die Option **–m** angeben, wie viele Verbindungen das RIM-Objekt maximal zum Konfigurations-Repository herstellen darf.

Anmerkung: Die Optionen **–a** und **–m** des Befehls **wcrtrim** werden momentan nur für RIM-Objekte von Tivoli Inventory verwendet. Aus diesem Grund werden diese Optionen nicht in der Dokumentation von Tivoli Management Framework beschrieben.

Um beispielsweise ein RIM-Objekt mit dem Namen invdh_2 zu erstellen und die maximale Anzahl von RDBMS-Verbindungen auf den Wert 10 zu setzen, müssen Sie folgenden Befehl ausführen:

```
wcrtrim -v Informix -h bernard -d inventory \
-H /data/rdbms/informix -u informix -s az914shm \
-a invdh -m 10 invdh_2
```

Dabei gilt:

–v Informix
: Gibt den Namen des Datenbanklieferanten an, für den das RIM-Objekt steht.

–h bernard
: Gibt den Namen des verwalteten Knotens an, auf dem sich das RIM-Objekt befindet.

–d inventory
: Gibt den Namen der Datenbank (Datenbank-ID) an, zu der das RIM-Objekt eine Verbindung herstellt.

–H /data/rdbms/informix
: Gibt den Pfad des Ausgangsverzeichnisses der Datenbank an.

–u informix
: Gibt den Namen des Datenbankbenutzers an, den das RIM-Objekt verwendet.

–s az914shm
: Gibt die Server-ID für die Datenbank an.

–a invdh
: Gibt an, dass die Anwendungskennung für das RIM-Objekt ’invdh’ lautet. Für RIM-Objekte, die vom Inventory Data Handler zum Herstellen einer Verbindung zum Verwaltungssystem für relationale Datenbanken verwendet werden, sollte dieser Wert auf ’invdh’ gesetzt werden.

–m 10 Gibt die maximale Anzahl der Verbindungen an, die zwischen dem RIM-Objekt und der Datenbank bestehen. Geben Sie für diese Option einen positiven Wert von 1 bis 16 ein. Der Standardwert ist 1. Es wird empfohlen, dass die Summe der RDBMS-Verbindungen, die für alle RIM-Objekte, die vom Inventory Data Handler verwendet werden, eingerichtet wurden, der Anzahl der Ausgabethreads entspricht, die für den Inventory Data Handler festgelegt wurden.

invdh_2
Gibt den Namen des RIM-Objekts an.

RIM-Objekte konfigurieren

In diesem Abschnitt wird erläutert, wie Sie die aktuellen Einstellungen für die RIM-Objekte, die vom Inventory Data Handler verwendet werden, anzeigen und ändern.

Sie können die meisten Optionen für RIM-Objekte mit den Befehlen **wgetrim** und **wsetrim** anzeigen und festlegen. Weitere Informationen zu diesen Befehlen können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen. Um die Werte für die Anwendungskennung und die maximale Anzahl der Datenbankverbindungen anzuzeigen und festzulegen, müssen Sie jedoch die folgenden Befehle verwenden. Stellen Sie zunächst die Objekt-ID (OID) des RIM-Objekts durch Ausgabe des folgenden Befehls fest:

```
wlookup RIM_Objektname
```

Dabei steht *RIM_Objektname* für den Namen des RIM-Objekts, z. B. invdh_1.

Die Ausgabe dieses Befehls entspricht in etwa dem folgenden Beispiel:

```
invdh_1 1228633823.2.20#RIM::RDBMS_Interface#
```

In diesem Beispiel ist 1228633823.2.20#RIM::RDBMS_Interface# die OID für das RIM-Objekt invdh_1. Führen Sie folgenden Befehl aus, um die Anwendungskennung für dieses RIM-Objekt anzuzeigen:

```
idlcall OID _get_application_type
```

Dabei steht *OID* für die OID des RIM-Objekts, die in diesem Beispiel wie folgt lautet: 1228633823.2.20#RIM::RDBMS_Interface#.

Für RIM-Objekte, die vom Inventory Data Handler verwendet werden, sollte dieser Wert auf invdh gesetzt sein. Für andere RIM-Objekte sollte dieser Wert auf Null ("") gesetzt sein. Geben Sie den folgenden Befehl ein, um diesen Wert zu ändern:

```
wsetrim -a Wert RIM_Objektname
```

Führen Sie den folgenden Befehl aus, um die maximale Anzahl der RDBMS-Verbindungen, die für dieses RIM-Objekt eingerichtet wurden, anzuzeigen:

```
idlcall OID _get_max_conn
```

Dabei steht *OID* für die OID des RIM-Objekts.

Geben Sie den folgenden Befehl ein, um diesen Wert zu ändern:

```
wsetrim -m 10 RIM_Objektname
```

Geben Sie für diese Option einen positiven Wert von 1 bis 16 ein. Der Standardwert ist 1.

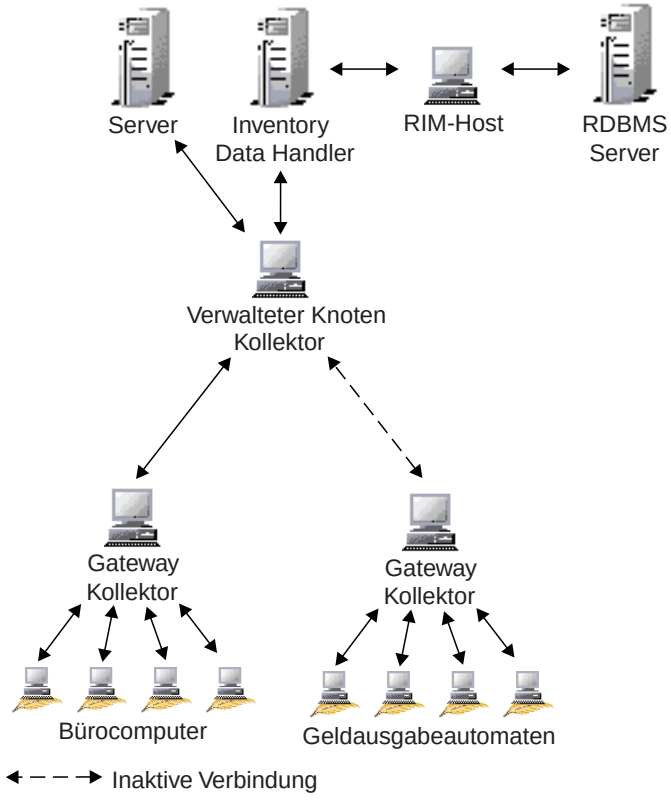
Weitere Informationen zu RIM-Objekten, die zusammen mit Tivoli Inventory installiert werden, finden Sie unter „Kennwörter für RIM-Objekte ändern” auf Seite 51.

Datenerfassung terminieren

Bei der Terminierung der Datenerfassung geben Sie an, dass Daten zu einem bestimmten Zeitpunkt von bestimmten Kollektoren der untergeordneten Ebene nicht erfasst werden sollen. Unter der Terminierung der Datenerfassung versteht man die Terminierung von *inaktiven Verbindungen*. Unter inaktiven Verbindungen versteht man Verbindungen zu Kollektoren, von denen Sie keine Daten abrufen möchten.

Anmerkung: Nur Verbindungen zwischen Kollektoren können deaktiviert werden. Verbindungen zwischen Kollektoren und Endpunkten können nicht deaktiviert werden.

Sie können den Zeitpunkt bestimmen, zu dem Daten durch die Datenerfassungs-Hierarchie fließen, indem Sie die Zeiten festlegen, in denen die Verbindungen zwischen den Kollektoren aktiv bzw. inaktiv sind. Eine Bank verfügt beispielsweise über ein Netzwerk, zu dem sowohl Bankautomaten als auch verschiedene andere Computer gehören. Da die Bankautomaten während der Geschäftszeiten schnell auf die auf dem Netzwerkserver gespeicherten Daten zugreifen müssen, können Abfragen an die Bankautomaten in diesen Zeiten von der Bank verhindert werden. Die folgende Abbildung soll dieses dieses Beispiel veranschaulichen.



Wenn Sie eine Verbindung zwischen Kollektoren deaktivieren, wird das CTOC für eine anstehende Datenerfassung in die Warteschlange für verzögerte Abfragen des Kollektors der untergeordneten Ebene gestellt. Bei Wiederherstellung der Verbindung zwischen den Kollektoren, wird das CTOC in der Warteschlange für verzögerte Abfragen verarbeitet.

Sie terminieren die Erfassung von Daten, indem Sie Tasks und Jobs in Tivoli Management Framework definieren und anschließend die Jobs unter Verwendung des Tivoli-Schedulers terminieren . Um dies tun zu können, müssen Sie mit den Konzepten, die im Kapitel über Task-Bibliotheken im *Tivoli Management Framework Benutzer-*

handbuch beschrieben werden, vertraut sein. Der folgende Abschnitt gibt eine Übersicht über die Prozedur zur Terminierung der Datenerfassung.

1. Erstellen Sie eine Task, mit der die Verbindungen zu einem Kollektor deaktiviert werden und mit der der Kollektor anschließend gestoppt und erneut gestartet wird, damit die Änderungen wirksam werden:

   ```
   wcollect -x offlinks_range_to_prohibit_collection Kollektor
   wcollect -h immediate
   wcollect -s
   ```

   Dabei gilt:

   offlinks_range_to_prohibit_collection
   : Gibt die Dispatcher-Nummern für das Objekt der Kollektoren an, deren Verbindungen zu dem angegebenen Kollektor deaktiviert werden müssen.

   Kollektor
   : Gibt den Kollektor an, dessen Verbindungen deaktiviert werden müssen.

   Sie können die Objekt-Dispatcher-Nummer eines Kollektors mit dem Befehl **odadmin** und der Option **odlist** abrufen. Weitere Informationen können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

   Sie können die Objekt-Dispatcher-Nummern wie im folgenden Beispiel in aufsteigender Reihenfolge und durch Kommas voneinander getrennt angeben:

   ```
   wcollect -x "4,5,6,7" Kollektor
   ```

   Sie können jedoch auch einen Bindestrich verwenden, um einen Bereich von Objekt-Dispatcher-Nummern anzugeben:

   ```
   wcollect -x "4-7" Kollektor
   ```

   Sie müssen den Bereich der inaktiven Verbindungen in doppelte Anführungsstriche (″″) setzen.

Weitere Informationen zum Befehl **wcollect** und der Option **–x** können Sie dem Abschnitt „wcollect” auf Seite 247 entnehmen.

2. Erstellen Sie eine Task, mit der die Verbindungen zu allen Systemen, deren Verbindungen vorher inaktiviert wurden, aktiviert werden und mit der der Kollektor anschließend gestoppt und erneut gestartet wird, damit die Änderungen wirksam werden:

   ```
   wcollect -x "" Kollektor
   wcollect -h immediate
   wcollect -s
   ```

3. Wiederholen Sie die Schritte 1 und 2 für alle Kollektoren, für die Sie den Abruf von Daten terminieren möchten.
4. Erstellen Sie Jobs für die Ausführung dieser Tasks.
5. Verwenden Sie den Tivoli-Scheduler, um zu steuern, wann und wie oft diese Jobs ausgeführt werden sollen.

Sie können die Erfassung von Daten ebenfalls terminieren, indem Sie einen Job erstellen, der Kollektoren über die Optionen **–s** und **–h** des Befehls **wcollect** startet und stoppt. Alternativ dazu können Sie einen Job terminieren, über den die Ausgabewarteschlange bzw. Eingabewarteschlange eines Kollektors deaktiviert wird, indem Sie die Option**–o** bzw. **–t** des Befehls **wcollect** für diesen Kollektor auf den Wert 0 setzen.

Anmerkung: Wenn Sie einen Kollektor rekonfiguriert haben, wird die Änderung erst dann wirksam, wenn Sie den Kollektor erneut starten.

Datenfluss im Netzwerk steuern

SCS bietet Funktionen zur Steuerung des SCS-Datenflusses in Ihrem Netzwerk. Diese Funktionen sind hilfreich, wenn Ihre Netzverbindungen langsam sind oder wenn Sie angeben möchten, zu welchem Zeitpunkt SCS-Daten durch Ihr Netzwerk fließen sollen. Zur Steuerung des Datenflusses stellt SCS folgende Mechanismen zur Verfügung:

- **Inaktive Verbindungen** - Inaktive Verbindungen sind die wichtigste Methode zur Steuerung des SCS-Datenflusses in Ihrem

Netzwerk. Mit Hilfe von inaktiven Verbindungen können Sie den SCS-Datenfluss zwischen Kollektoren zu bestimmten Zeiten aktivieren und deaktivieren. Um Verbindungen zu deaktivieren, müssen Sie einen Kollektor an beiden Enden des Netzwerks, für das der Datenfluss gesteuert werden soll, installieren und die Deaktivierung der Verbindungen mit Hilfe des Tivoli-Schedulers terminieren. Weitere Informationen zum Terminieren von inaktiven Verbindungen können Sie den Abschnitten „Datenerfassung terminieren" auf Seite 109 und „wcollect" auf Seite 247 entnehmen.

- **Größe der Übertragungsblöcke** - Mit der Option **–c** des Befehls **wcollect** können Sie die Größe der Übertragungsblöcke konfigurieren. Über diese Option können Sie die Größe der SCS-Datenpakete steuern, die die IOM-Verbindung (Interobject message) passieren. Der Standardwert für die Größe des Übertragungsblocks beträgt 1 MB. Wenn Sie eine kleinere Größe angeben, werden die Anwendungsdaten in kleineren Blöcken gesendet. Bei langsamen Verbindungen empfiehlt es sich, die Übertragungsblöcke zu verkleinern. Dadurch kann vermieden werden, dass sich große SCS-Datenblöcke in den Verbindungen stauen. Die Größe der Übertragungsblöcke wird auf dem Kollektor der untergeordneten Ebene konfiguriert. Weitere Informationen zum Konfigurieren der Größe von Übertragungsblöcken können Sie den Abschnitten „Kollektoren konfigurieren" auf Seite 96 und „wcollect" auf Seite 247 entnehmen.

  Anmerkung: Die Deaktivierung von Verbindungen und die Größe von Übertragungsblöcken wirken sich auf die Datenübertragung zwischen Kollektoren bzw. zwischen einem Kollektor und dem Inventory Data Handler aus. Datenübertragungen zwischen Endpunkten und Gateway-Kollektoren sind davon nicht betroffen.

- **Eingabethreads** - Kollektoren verwenden Eingabethreads, um eine IOM-Sitzung mit einem Kollektor der untergeordneten Ebene zum Abrufen von Daten zu starten. Sie können über die Option **–t** des Befehls **wcollect** die maximale Anzahl von Eingabethreads für einen Kollektor festlegen. Sie können die maximal

zulässige Anzahl von Eingabethreads erhöhen, damit mehr SCS-IOM-Sitzungen gleichzeitig ablaufen können. Sie können aber auch die Anzahl herabsetzen, um die SCS-IOM-Daten, die in den Kollektor fließen, zu reduzieren. Weitere Informationen zum Konfigurieren der Eingabethreads eines Kollektors finden Sie unter „Threads konfigurieren” auf Seite 98 und „wcollect” auf Seite 247.

Anmerkung: Durch die Konfiguration der Eingabethreads eines Gateway-Kollektors können Sie die Gateway-Auslastung verringern, indem Sie die Anzahl der gleichzeitig ablaufenden SCS-Sitzungen zwischen dem Gateway und Endpunkten reduzieren. Auf diese Weise können Sie auch die Anzahl der SCS-Netzdaten verringern, die an das Gateway gesendet werden.

Informationen zu SCS abrufen

Mit Hilfe verschiedener SCS-Befehle können Sie die folgenden Informationen anzeigen:

- Die Konfiguration und den Status eines Kollektors
- Eine Beschreibung der CTOCs auf einem Kollektor
- Den Inhalt der Warteschlangen eines Kollektors
- Die Konfiguration des Inventory Data Handler
- Informationen zu aktuellen Inventory-Suchläufen
- Informationen zu Optionen für SCS- und Tivoli Inventory-Statuskollektoren eines Profils

In diesem Abschnitt finden Sie eine Beschreibung der Befehle zum Abrufen von Informationen über SCS sowie Informationen zum Anzeigen der im Konfigurations-Repository gespeicherten Inventory-Daten. Weitere Informationen zur Verwendung dieser Befehle können Sie dem Abschnitt „Befehle” auf Seite 239 entnehmen.

Kollektor-Konfiguration

Mit dem Befehl **wcollect** können Sie Informationen zur aktuellen Konfiguration eines Kollektors abrufen. Sie können die Stufe der Debug-Informationen, die in die Protokolldatei des Kollektors geschrieben werden, sowie die maximale Größe der Protokolldatei, die Adresse des Laufzeitverzeichnisses, die Depotgröße, die Größe der Übertragungsblöcke, die Thread-Einstellungen, die Einstellungen für Wiederholungen des Datenabrufs, inaktive Verbindungen und andere Informationen anzeigen.

Geben Sie den folgenden Befehl ein, um Konfigurationsdaten anzuzeigen:

```
wcollect Kollektor
```

Dabei hat *Kollektor* das folgende Format: **@ManagedNode:***Kollektor*, **@Gateway:***Kollektor* bzw. **@InvDataHandler:Inventory Data Handler**.

Kollektorstatus

Mit dem Befehl **wcstat** können Sie Statusinformationen zu einem Kollektor abrufen. Diese Informationen sind für die Behebung von Fehlern, die auf Kollektoren aufgetreten sind, hilfreich.

Geben Sie den folgenden Befehl ein, um Statusinformationen zu einem Kollektor anzuzeigen:

```
wcstat Kollektor
```

Dabei hat *Kollektor* das folgende Format: **@ManagedNode:***Kollektor* bzw. **@Gateway:***Kollektor*.

CTOCs und Warteschlangen

Mit dem Befehl **wcstat** können Sie Informationen zu CTOCs oder zu den Warteschlangen eines Kollektors abrufen.

Informationen zu CTOCs enthalten folgende Angaben: die CTOC-ID, Priorität der Suchlaufdaten, Kollektorstatus, Quelle und Zieladresse der Suchlaufdaten, die für das Zielobjekt aufzurufende Methode,

Client-Merkmale, Status der Datenerfassung, Anzahl der Wiederholungen. Geben Sie den folgenden Befehl ein, um Informationen zu CTOCs anzuzeigen:

```
wcstat -v Ctoc_ID Kollektor
```

Dabei gibt *Ctoc_ID* das CTOC an, für das Sie Statusinformationen anfordern. *Kollektor* muss folgendes Format haben: **@ManagedNode:***Kollektor* bzw. **@Gateway:***Kollektor*.

Kollektoren verfügen über Eingabe-, Ausgabe-, Fehlerwarteschlangen und Warteschlangen für beendete und verzögerte Datenerfassungsvorgänge. Zu all diesen Warteschlangenarten können Sie folgende Informationen anzeigen: CTOC-ID, Priorität der Daten in den Warteschlangen, Kollektorstatus, Informationen zur Quelle und zur Zieladresse, die für das Zielobjekt aufzurufende Methode, Suchlauf-ID, Status der Datenerfassung und die Anzahl der Versuche, die dieser Kollektor unternommen hat, um Anforderungen an einen Kollektor der übergeordneten Ebene zu senden.

Geben Sie den folgenden Befehl ein, um Informationen zu den Warteschlangen eines Kollektors anzuzeigen:

```
wcstat -q [ioecd] Kollektor
```

Dabei steht **[ioecd]** für die Warteschlange, für die Sie Informationen anfordern (Eingabe-, Ausgabe-, Fehlerwarteschlange, beendete und/oder verzögerte Warteschlange). *Kollektor* muss folgendes Format haben: **@ManagedNode:** *Kollektor* bzw. **@Gateway:** *Kollektor*.

Inventory Data Handler-Konfiguration

Mit dem Befehl **wgetinvdh** können Sie Konfigurationsdaten zum Inventory Data Handler abrufen, einschließlich Benachrichtigungen über die Beendigung von Suchläufen, Anzahl von Wiederholungen und Zeitlimit zum Schreiben von Daten in RIM-Objekte. Über die Option **–a** des Befehls **wgetinvdh** können Sie alle verfügbaren Informationen zum Inventory Data Handler abrufen. Über eine Reihe von anderen Optionen können Sie selektiv Informationen anzeigen. Weitere Informationen hierzu können Sie dem Abschnitt „wgetinvdh” auf Seite 274 entnehmen.

Inventory-Suchläufe

Mit dem Befehl **wgetscanstat** können Sie Informationen zu Inventory-Suchläufen anzeigen. Wenn Sie den Befehl **wgetscanstat** ohne Angabe von Optionen ausführen, erhalten Sie eine Liste mit den IDs der gegenwärtig anstehenden Inventory-Suchläufe und den Namen der Inventory-Profile, die diese Suchläufe eingeleitet haben. Sie können jedoch auch Optionen angeben, über die Informationen zum Status (erfolgreich ausgeführt, fehlgeschlagen bzw. anstehend) bestimmter Suchläufe bzw. aller aktuellen Suchläufe angezeigt werden. Weitere Informationen können Sie dem Abschnitt „wgetscanstat" auf Seite 306 entnehmen.

Inventory-Profile

Mit dem Befehl **wgetinvglobal** können Sie Informationen zu den Einstellungen für Statusbenachrichtigungen, Verteilungen und Daten für Inventory-Profile abrufen. Wenn Sie bei Ausführung dieses Befehls keine Optionen angeben, werden alle verfügbaren Informationen zu diesen Einstellungen für das Inventory-Profil angezeigt. Über die Angabe von anderen Optionen können Sie Informationen selektiv anzeigen. Weitere Informationen können Sie dem Abschnitt„wgetinvglobal" auf Seite 277 entnehmen.

Im Konfigurations-Repository gespeicherte Daten

Um die im Konfigurations-Repository gespeicherten Daten anzuzeigen, können Sie die mit Tivoli Inventory gelieferten vordefinierten Abfragen verwenden oder eigene Abfragen erstellen. Weitere Informationen können Sie dem Abschnitt „Inventory-Daten abfragen" auf Seite 209 entnehmen.

Statusinformationen in Tivoli Inventory konfigurieren

Sie können Tivoli Inventory so konfigurieren, dass Statusinformationen zu Suchläufen zurückgegeben werden. Der Begriff Status bezieht sich auf die erfolgreiche bzw. fehlgeschlagene Ausführung eines Suchlaufs auf einem bestimmten Ziel eines Inventory-Profils.

Mit den Befehlen **wsetinvdh** und **wsetinvglobal** können Sie Optionen für Benachrichtigungen und die Protokollierung von Statusinformationen in Tivoli Inventory konfigurieren:

- Über die Option **–s** des Befehls **wsetinvdh** können Sie angeben, ob der Inventory Data Handler Statusinformationen speichert, die im Falle eines Systemabsturzes wiederhergestellt werden können.
- Über die Option **–f** des Befehls **wsetinvglobal** können Sie den Namen der Protokolldatei angeben, in der die Statusinformationen gespeichert werden. Über die Option **–h** des Befehls **wsetinvglobal** können Sie den Host angeben, auf dem die Protokolldatei gespeichert wird. Diese Optionen sind nur verfügbar, wenn Sie über die Option **–l** des Befehls **wsetinvglobal** festgelegt haben, dass Benachrichtigungen an eine Protokolldatei gesendet werden.
- Über die Option**–n** des Befehls **wsetinvglobal** können Sie den Zeitpunkt festlegen, zu dem Benachrichtigungen gesendet werden. Sie haben folgende Optionen: die Benachrichtigungen sofort, in regelmäßigen Abständen oder bei Beendigung der Suchläufe auf allen Zielen senden.
- Über die Option **–t** des Befehls **wsetinvglobal** können Sie festlegen, in welchen Fällen eine Benachrichtigung für das angegebene Profil gesendet wird. Sie haben folgende Optionen: die Benachrichtigung senden, wenn die Suchläufe erfolgreich beendet werden konnten, die Suchläufe fehlgeschlagen sind oder wenn die Suchläufe auf allen Zielen beendet sind.
- Über die Option **–l** des Befehls **wsetinvglobal** können Sie angeben, an welche Zieladresse die Benachrichtigungen gesendet werden. Sie haben folgende Optionen: die Benachrichtigung an die Benachrichtigungsgruppe von Tivoli Inventory bzw. an die Protokolldatei und/oder die TEC-Konsole senden. Es besteht auch die Möglichkeit, keine Benachrichtigungen zu senden.

Mit den Optionen **–n** und **–q** des Befehls **wcrtinvdh** bzw. **wsetinvdh** und der Option **–n** des Befehls **wsetinvglobal** können Sie die Art der Bündelung von Statusinformationen festlegen.

Mit dem Befehl **wgetscanstat** können Sie Statusinformationen zu aktuellen Inventory-Suchläufen abrufen. Sie können mit dem Befehl **wgetinvglobal** Informationen zu Statuskollektoroptionen für ein bestimmtes Profil abrufen.

Weitere Informationen zu diesen Befehlen und den zugehörigen Optionen können Sie dem Abschnitt „Befehle" auf Seite 239 entnehmen.

Hinweise zu miteinander verbundenen TMRs

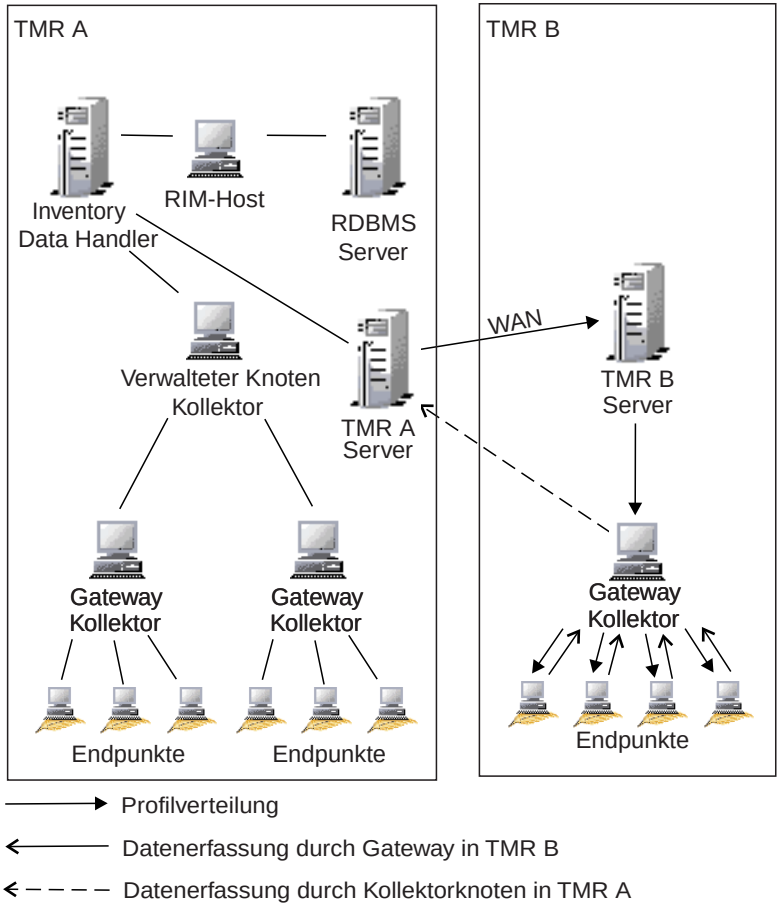
Sie können TMRs unter Verwendung einer Einweg- oder Zweiwegeverbindung miteinander verbinden. Um SCS in mehreren verschiedenen TMRs zu verwenden, müssen Sie eine Zweiwegeverbindung zwischen den TMRs konfigurieren.

Wenn Sie TMRs miteinander verbinden, müssen Sie die Ressourcen aktualisieren. Wenn TMRs miteinander verbunden werden, werden zunächst die Kennungen für Namen und Objekte, die sich in der Namensregistrierdatenbank des jeweiligen Servers befinden, ausgetauscht. Diese werden dann in regelmäßigen Abständen aktualisiert . Weitere Informationen zum Verbinden von TMRs und Aktualisieren von Ressourcen können Sie dem Handbuch *Tivoli Management Framework: Einsatzplanung* entnehmen.

Wenn Sie Suchläufe in zwei verschiedenen TMRs durchführen möchten und Tivoli Inventory die Daten unter Verwendung von SCS zurückgeben soll, müssen Sie SCS in beiden TMRs installieren.

SCS verwendet MDist 2-Repeater-Hierarchien zum Abrufen von WAN-Eingangspunktinformationen. Beispiel: Sie verfügen über zwei TMRs, die über ein WAN miteinander verbunden sind: TMR A und TMR B. Der Inventory Data Handler befindet sich in TMR A. Ein Profil verteilt Daten von TMR A an Endpunkte in TMR B. Der Gateway für die Endpunkte in TMR B ruft die Ergebnisse des Suchlaufs von diesen Endpunkten ab. Anschließend ruft der Kollektorknoten des WAN-Eingangspunktes in TMR A die Ergebnisse des Suchlaufs vom Gateway in TMR B ab und sendet die Daten an den Inventory Data Handler.

In der folgenden Abbildung werden diese Prozesse veranschaulicht.



Weitere Vorgehensweise

Nach der Konfiguration von SCS können Sie Daten über die von Ihnen verwalteten Systeme erfassen und speichern. Dazu müssen Sie Inventory-Profile erstellen. Weitere Informationen können Sie dem Abschnitt „Mit Inventory-Profilen arbeiten” auf Seite 163 entnehmen.

5

Benutzerspezifische Daten mit Tivoli Inventory erfassen

Dieses Kapitel enthält Informationen zur Vorgehensweise bei der Erfassung benutzerspezifischer und optionaler Daten mit Hilfe von Tivoli Inventory. Sie haben die Möglichkeit, benutzerspezifische Daten durch Erstellen von Softwaresignaturen im Konfigurations-Repository zu erfassen. Darüber hinaus können Sie benutzerspezifische Tabellen hinzufügen und mit Hilfe von Tivoli UserLink benutzerspezifische Daten erfassen, mit denen diese Tabellen ausgefüllt werden. In diesem Kapitel wird darüber hinaus auch erläutert, wie Benutzer Ihnen benutzerspezifische Daten zur Verfügung stellen können.

Durch einige zusätzliche Konfigurationseinstellungen können Sie die Funktion zur Verteilung von Inventory-Profilen für weitere Zwecke einsetzen. Bei einer Anpassung der Konfiguration wird jedoch vorausgesetzt, dass Sie mit den Optionen des Inventory-Profils vertraut sind. Anweisungen zur Anpassung von Inventory-Profilen können Sie dem Kapitel „Mit Inventory-Profilen arbeiten” auf Seite 163 entnehmen.

Bei der Anpassung eines Inventory-Profils können Sie die Art des Suchlaufs auf den Zielmaschinen festlegen und angeben, welche Daten gespeichert werden sollen. Neben der Erstellung von Inventory-Profilen bietet Tivoli Inventory außerdem die Möglichkeit, benutzerspezifische Daten zu erfassen: für Administratoren über die

Web-basierte Schnittstelle von Tivoli Inventory und für Benutzer über die Web-basierte Schnittstelle UserLink for Tivoli Inventory.

Die Web-basierte Schnittstelle für Administratoren stellt eine Benutzerdatenschablone zur Verfügung, ein Web-Formular, über das der Administrator die vom Benutzer zur Verfügung gestellten benutzerspezifischen Daten angeben kann. Die Benutzerdatenschablone gibt außerdem die MIF-Gruppen und -Attribute an, die die Datei useradd.mif, die von UserLink for Tivoli Inventory verwendet wird, enthalten soll. Die in der Benutzerdatenschablone angegebenen Attribute sind im Benutzerdatenformular als Textfenster dargestellt; bei dem Benutzerdatenformular handelt es sich um ein Web-Formular, in dem Benutzer benutzerspezifische Daten eingeben können. Weitere Informationen hierzu finden Sie unter „Benutzerdaten erfassen” auf Seite 126.

Die Web-basierte Schnittstelle für Benutzer, UserLink for Tivoli Inventory, stellt ein Benutzerdatenformular zur Verfügung. Hierbei handelt es sich um ein Web-Formular, auf das Zugriff über UserLink besteht; in diesem Formular können Benutzer Informationen zu der eigenen Maschine eingeben, die während eines Hardware- bzw. Softwaresuchlaufs nicht erfasst wurden. Wenn der Benutzer in diesem Formular auf ’Übergeben’ klickt, wird die Datei useradd.mif erstellt. Weitere Informationen hierzu finden Sie unter „MIF-Benutzerdateien verwenden” auf Seite 158.

Mit Softwaresignaturen arbeiten

Bei Softwaresignaturen handelt es sich um Daten zur Identifikation von Softwarepaketen wie beispielsweise den Namen und die Größe der ausführbaren Datei für das Softwarepaket. Die in Tivoli Inventory standardmäßig zur Verfügung gestellten Softwaresignaturen werden bei der Installation in das Konfigurations-Repository geladen. Siehe hierzu „Softwaresignaturen installieren” auf Seite 85. Diese Softwaresignaturen können Sie editieren bzw. löschen, oder Sie können eigene Softwaresignaturen hinzufügen. Beispielsweise können Sie eine Softwaresignatur für eine hausintern entwickelte Anwendung hinzufügen.

Anhand von Softwaresignaturen kann von Tivoli Inventory ermittelt werden, welche Softwarepakete sich auf den Maschinen befinden, auf denen Sie Suchläufe ausführen. Wenn Sie einen Softwaresignaturen-Suchlauf auf einem Endpunkt durchführen, werden die Softwaresignaturen von Tivoli Inventory an den Endpunkt verteilt; anschließend werden alle Dateien auf dem Endpunkt mit der Liste der Softwaresignaturen verglichen. Wenn eine Datei einer Softwaresignatur entspricht, werden die Daten der Softwaresignatur für diese Datei an das Konfigurations-Repository gesendet. Da nur Daten für Dateien, die den Suchkriterien entsprechen, gesendet werden, werden bei diesem Suchlauf weniger Daten an das Konfigurations-Repository zurückgegeben als das bei Suchläufen für allgemeine Dateidaten oder Header-Informationen der Fall ist.

Softwaresignaturdaten beinhalten den Namen, die Größe und normalerweise den Kontrollsummenwert der Datei, die zur Identifizierung des Softwareprodukts verwendet wird. Diese zur Identifizierung verwendete Datei ist normalerweise die primäre ausführbare Datei für das Produkt wie beispielsweise die Datei notepad.exe. Die während eines Suchlaufs erfassten Softwaresignaturdaten werden im Konfigurations-Repository in der Tabelle MATCHED_SWARE gespeichert. Unter Verwendung der Abfrage INVENTORY_SWARE können Sie sämtliche Softwareinformationen zu einer Maschine zurückgeben.

Bei Softwaresignatur-Suchläufen werden nur Daten für Dateien, die den Suchkriterien entsprechen, zurückgegeben. Wenn Sie in Ihrer Firma ausschließlich Softwaresignatur-Suchläufe durchführen, enthält das Konfigurations-Repository nur Daten für Dateien, die den Suchkriterien entsprechen, und nicht für alle Dateien, die sich auf den jeweiligen Systemen befinden. Wenn Sie eine neue Softwaresignatur erstellt oder eine vorhandene Softwaresignatur editiert haben, müssen Sie erneut Suchläufe durchführen, um die neuen bzw. überarbeiteten Softwaresignaturen zu erfassen.

Wenn Sie jedoch einen Suchlauf zur Erfassung von allgemeinen Daten durchgeführt haben, enthält das Konfigurations-Repository Informationen zu allen Dateien auf den durchsuchten Systemen und nicht nur zu den Dateien, deren Daten den Softwaresignaturen entsprechen. In diesem Fall können Sie eine Abfrage absetzen, um die gewünschten Dateidaten anzuzeigen, anstatt die Softwaresignaturen zu ändern und erneut Suchläufe durchzuführen. Wenn Sie Abfragen zu allgemeinen Informationen an das Konfigurations-Repository absetzen möchten, gehen Sie so vor wie in Tivoli Inventory Version 3.6.*x*.

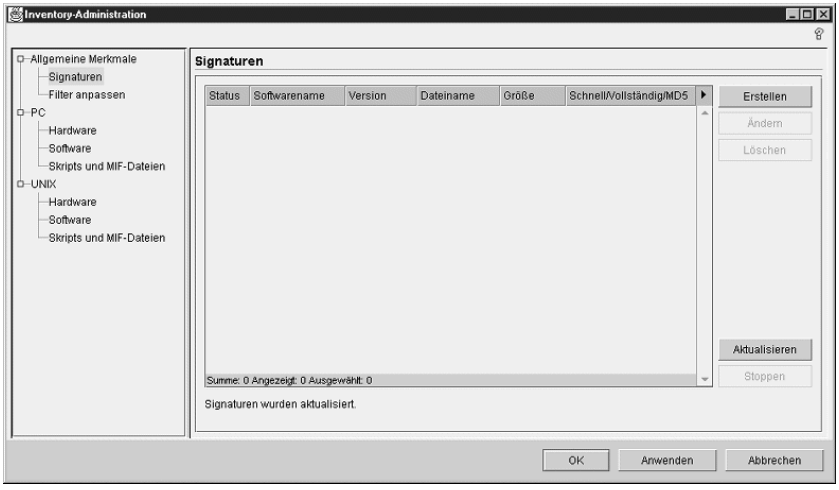
Sie können im Fenster 'Signaturen' oder mit dem Befehl **winvsig** Softwaresignaturen im Konfigurations-Repository hinzufügen, editieren oder löschen.

In der folgenden Tabelle sind die zur Ausführung dieser Task erforderlichen Berechtigungsklassen aufgeführt:

Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Softwaresignaturen im Konfigurations-Repository hinzufügen, editieren oder löschen	Das Fenster 'Signaturen'	**RIM_view**, **RIM_update**, **senior** oder **super**

Über die Arbeitsoberfläche

Um das Fenster 'Signaturen' zu öffnen, klicken Sie auf das Feld neben **Globale Merkmale** im linken Teilbereich des Fensters 'Inventory-Verwaltung', um die Komprimierung der Anzeige 'Globale Merkmale' aufzuheben, falls dies notwendig ist, und das Fenster **Signaturen** aufzurufen. Das Fenster 'Signaturen' wird angezeigt.



Weitere Informationen zum Hinzufügen, Editieren oder Löschen von Softwaresignaturen im Konfigurations-Repository unter Verwendung des Fensters 'Signaturen' können Sie der Tivoli Inventory Online-hilfe entnehmen.

Über die Befehlszeile

Sie können Softwaresignaturen mit dem Befehl **winvsig** hinzufügen oder löschen. Soll beispielsweise dem Konfigurations-Repository eine Softwaresignatur für das Softwarepaket Microsoft Office 2000 hinzugefügt werden, müssen Sie diesen Befehl eingeben:

```
winvsig –a –n MSOFFICE.EXE –s 405560 \
–d "Microsoft Office 2000" –v "9.0.2601"
```

Dabei gilt:

–a Gibt an, dass dem Konfigurations-Repository eine Softwaresignatur hinzugefügt werden soll.

–n MSOFFICE.EXE
Gibt den Namen der Datei an, die zur Identifizierung des Softwarepakets verwendet wird.

–s 405560
Gibt die Größe der Datei MSOFFICE.EXE (in Byte) an.

–d “Microsoft Office 2000”
Gibt die Bezeichnung der Software an. Dies entspricht dem Feld ’Softwarename’ in den Dialogen ’Signatur editieren’ und ’Neue Signatur’.

–v “9.0.2601”
Gibt die Version der Software an, für die diese Signatur steht.

Im folgenden Beispiel soll die Softwaresignatur für die Software Microsoft Word 97 aus dem Konfigurations-Repository entfernt werden:

```
winvsig –r –n WINWORD.EXE –s 5317904
```

Dabei gilt:

–r Gibt an, dass eine Softwaresignatur aus dem Konfigurations-Repository entfernt werden soll.

–n WINWORD.EXE
Gibt den Namen der Datei zur Identifizierung der zu löschenden Softwaresignatur an.

–s 5317904
Gibt die Größe der Datei WINWORD.EXE (in Byte) an.

Weitere Informationen zum Befehl **winvsig** können Sie dem Abschnitt „Befehle” auf Seite 239 entnehmen.

Benutzerdaten erfassen

Die Erfassung von Benutzerdaten mit Tivoli Inventory verläuft in zwei Schritten und erfordert Administrator- und Benutzerzugriff auf die Web-Seiten von Tivoli Inventory. Dabei wird vom Administrator zunächst eine Formularsammlung, die Benutzerdatenschablone, definiert. Im nächsten Schritt füllen Benutzer das Benutzerdatenformular aus, bei dem es sich um ein Web-Formular handelt, auf das Benutzer über UserLink for Tivoli Inventory zugreifen können.

In der Benutzerdatenschablone können Sie als Administrator die Felder für das Benutzerdatenformular festlegen. Durch Hinzufügen oder

Löschen von MIF-Gruppen bzw. -Attributen in der Benutzerdatenschablone legen Sie indirekt die MIF-Gruppen und -Attribute fest, die die MIF-Datei useradd.mif der Benutzerdatenschablone bei ihrer nächsten Erstellung auf den Zielmaschinen enthalten wird. Die von Ihnen erstellten Gruppen und Attribute werden in der Tivoli-Namensregistrierdatenbank gespeichert.

Bei jedem Aufruf des Benutzerdatenformulars über UserLink for Tivoli Inventory wird dieses Formular dynamisch anhand der in der Namensregistrierdatenbank vorliegenden aktuellen Gruppen und Attribute als Web-Seite erstellt. Das Benutzerdatenformular enthält ein Feld für jedes Attribut. (Informationen zur Namensregistrierdatenbank können Sie dem Abschnitt „Registrierte Namen" auf Seite 241 entnehmen.)

Wenn der Benutzer im Benutzerdatenformular auf 'Übergeben' klickt, wird die Datei 'useradd.mif' erstellt und lokal gespeichert. Um die in der Datei 'useradd.mif' enthaltenen Daten im Konfigurations-Repository zu speichern, müssen Sie ein Inventory-Profil so anpassen, dass die Datei 'useradd.mif' während der Profilverteilung gelesen wird. Dieser Vorgang wird über die grafische Benutzerschnittstelle (GUI) von Tivoli Inventory im Fenster 'Skript und MIF-Dateien' ausgeführt.

Bei der Übernahme der Daten in das Konfigurations-Repository werden diese in den Tabellen und Spalten gespeichert, die von Ihnen entsprechend der MIF-Gruppen und -Attribute in der Datei 'useradd.mif' erstellt wurden.

Übersicht

In den nachfolgenden Abschnitten wird anhand von Beispielen veranschaulicht, wie Sie den Vor- und Nachnamen, die Personal- und Raumnummer der einzelnen Benutzer abrufen und diese Angaben jeweils der Maschine zuordnen, auf der sie eingegeben wurden. Es folgt eine Zusammenfassung der Prozeduren, die für die Verwendung der Web-Schnittstelle von Tivoli Inventory zur Erfassung von benutzerdefinierten Daten erforderlich sind.

1. Legen Sie die Benutzerdaten, die abgerufen werden sollen, und die Ablage dieser Informationen im Konfigurations-Repository-Schema fest. Siehe hierzu „Erfassung von Benutzerdaten vorbereiten” auf Seite 129.
2. Legen Sie die Endbenutzer fest, denen die Zugriffsberechtigung auf UserLink for Tivoli Inventory erteilt werden soll. Richten Sie außerdem einen oder mehrere Tivoli-Administratoren mit Zugriffsberechtigung auf UserLink ein. Siehe hierzu „Benutzerzugriff auf UserLink erteilen” auf Seite 129.
3. Erstellen Sie neue Tabellen im Konfigurations-Repository, in denen die Benutzerdaten gespeichert werden sollen. (Benutzerdaten dürfen nicht in den standardmäßig vorhandenen Tabellen des Konfigurations-Repositorys abgelegt werden.) Siehe hierzu „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen” auf Seite 141.
4. Erstellen Sie mit Hilfe des Web-Formulars ’Benutzerdatenschablone’ die entsprechenden MIF-Gruppen und -Attribute. Durch diesen Vorgang werden die Felder im Benutzerdatenformular festgelegt, dem Web-Formular, das von den Benutzern ausgefüllt wird. Siehe hierzu „MIF-Gruppen und -Attribute anhand der Benutzerdatenschablone erstellen” auf Seite 143.
5. Weisen Sie die Benutzer an, eine Verbindung zu UserLink for Tivoli Inventory unter Verwendung der Datei userlink.htm herzustellen. Siehe hierzu „Datei userlink.htm verwenden” auf Seite 152. Die Benutzer sollten dann das Benutzerdatenformular ausfüllen. Wenn die Benutzer auf ’Übergeben’ klicken, wird die Datei useradd.mif erstellt. Siehe hierzu „Benutzerdatenformular ausfüllen” auf Seite 153.
6. Verteilen Sie über die Tivoli-Arbeitsoberfläche ein Inventory-Profil, um die Datei useradd.mif abzurufen. Siehe hierzu „Benutzerdaten abrufen und speichern” auf Seite 156.
7. Anschließend können Sie die Benutzerdaten abrufen, indem Sie eine neue Anzeige im Konfigurations-Repository erstellen. Siehe hierzu „Benutzerdaten anzeigen” auf Seite 156.

Erfassung von Benutzerdaten vorbereiten

Über UserLink for Tivoli Inventory können Sie alle vom Benutzer eingegebenen Daten abrufen. Eine ganze Reihe von Benutzerdaten werden bei einem Hardware- oder Softwaresuchlauf nicht erfasst; dazu gehören Geräteinformationen, beispielsweise zu Telefon oder Faxgerät. Diese konstanten Angaben lassen sich jedoch relativ einfach in einer MIF-Benutzerdatei erfassen, die von Ihnen einmal manuell erstellt wird. (Weitere Informationen hierzu können Sie dem Abschnitt „MIF-Benutzerdateien verwenden” auf Seite 158 entnehmen.) UserLink for Tivoli eignet sich am besten zur Erfassung dynamischer Informationen (z. B. über den aktuellen Standort einer Maschine, einschließlich Raum- und Gebäudenummer sowie Abteilungsnummer des Benutzers), da Sie die Benutzer zu einer regelmäßigen Aktualisierung des Benutzerdatenformulars anhalten können.

Benutzerzugriff auf UserLink erteilen

Über UserLink for Tivoli Inventory können Benutzer das Benutzerdatenformular ausfüllen. Lesen Sie diesen Abschnitt, wenn einige oder alle Benutzer die Möglichkeit erhalten sollen, diese Task auszuführen.

Benutzer, die Zugriff auf UserLink-Funktionen erhalten sollen, müssen in Tivoli Management Framework als Administratoren eingerichtet werden und die TMR-Berechtigungsklasse **Inventory_end_user** erhalten.

Sie haben die Möglichkeit, für jeden Benutzer einen Tivoli-Administrator zu erstellen. Es wird jedoch empfohlen, einen einzigen Administrator für mehrere Benutzer einzurichten, indem Sie im Dialogfenster ’Administrator erstellen’ den TMR-Server-Anmeldenamen der einzelnen Benutzer hinzufügen. Sie können außerdem einen Administrator erstellen, der über ein globales Benutzer-Account für den TMR-Server verfügt, und anschließend den Namen und das Kennwort allen Benutzern zugänglich machen, für die eine Zugriffsberechtigung auf UserLink erforderlich ist. (Stellen Sie sicher, dass alle Systemverzeichnisse sowie Verzeichnisse mit den TMR-

Installationsdateien schreibgeschützt sind; dadurch wird vermieden dass kritische Dateien auf dem TMR-Server von Endbenutzern geändert werden.)

Anmerkung: Die Berechtigungsklasse **Inventory_end_user** ist für Benutzer, die in der TMR über die Berechtigungsklasse **senior**, **super**, **user** oder **admin** verfügen, für den Zugriff auf UserLink nicht erforderlich.

Die Aktivierung der Links auf der Hauptseite von UserLink for Tivoli Inventory ist nur bei Eingabe des Anmeldenamens und des Kennworts für den TMR-Server möglich. Dies ermöglicht den Benutzern, eine sichere Verbindung zum TMR-Server herzustellen und auf die im Web verfügbaren Tivoli Inventory-Funktionen zuzugreifen.

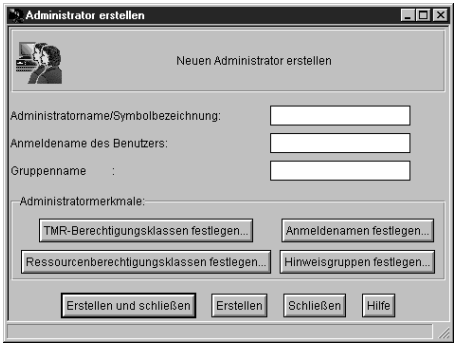
Derzeit können sowohl in Tivoli Inventory als auch in Tivoli Software Distribution bestimmte Funktionen über UserLink aufgerufen werden. Sind beide Anwendungen in Ihrer Tivoli-Umgebung installiert, können Sie Benutzern über die Web-Schnittstelle Zugriff auf beide Anwendungen erteilen, indem Sie dem Administrator die Berechtigungsklassen **Inventory_end_user** und **SWDist_end_user** zuweisen. Es besteht jedoch die Möglichkeit, Benutzern von UserLink for Tivoli Inventory und Benutzern von UserLink for Tivoli Software Distribution jeweils eine eigene Zugriffsberechtigung zu erteilen, indem Sie für jede Anwendung eigene Administratoren erstellen. Weitere Informationen zu UserLink for Tivoli Software Distribution können Sie dem *Tivoli Software Distribution Benutzerhandbuch* entnehmen.

Sie können Administratoren für den Zugriff auf UserLink über die Arbeitsoberfläche oder über die Befehlszeile erstellen. Weitere Informationen zum Erstellen von Administratoren können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

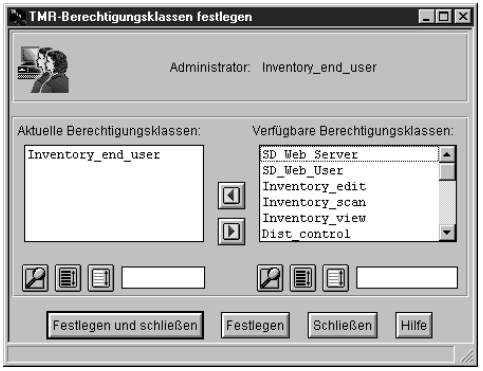
Über die Arbeitsoberfläche

Folgende Schritte sind zur Erteilung des Benutzerzugriffs auf User-Link erforderlich:

1. Starten Sie die Tivoli-Arbeitsoberfläche auf dem TMR-Server.
2. Doppelklicken Sie auf das Symbol **Administratoren**, um das Fenster 'Administratoren' zu öffnen.
3. Wählen Sie **Erstellen->Administrator** aus. Daraufhin wird das Dialogfenster 'Administrator erstellen' geöffnet.



4. Geben Sie im Feld **Administratorname/Symbolbezeichnung** die entsprechenden Informationen ein.
5. Geben Sie im Feld **Anmeldename des Benutzers** den Anmeldenamen für den TMR-Server eines Benutzers ein, der als Administrator Zugriff auf UserLink erhalten soll.
6. (Optional) Geben Sie im Feld **Gruppenname** einen Gruppennamen ein. Weitere Informationen zu Gruppen können Sie dem Abschnitt zu Tivoli-Administratoren im *Tivoli Management Framework Benutzerhandbuch* entnehmen.
7. Klicken Sie auf **TMR-Berechtigungsklassen festlegen**. Daraufhin wird das Dialogfenster 'TMR-Berechtigungsklassen festlegen' geöffnet. Gehen Sie in diesem Dialogfenster wie folgt vor:



a. Wählen Sie in der Liste **Verfügbare Berechtigungsklassen** den Eintrag **Inventory_end_user** aus. Sie sollten keine weiteren Tivoli-Berechtigungsklassen auswählen, um die Sicherheit Ihrer Tivoli-Umgebung aufrechtzuerhalten.

b. Klicken Sie auf den Linkspfeil; dadurch wird der ausgewählte Eintrag in die Liste **Aktuelle Berechtigungsklassen** verschoben.

c. Klicken Sie anschließend auf **Festlegen und schließen**, um Ihre Änderungen zu sichern und in den Dialog 'Administrator erstellen' zurückzukehren.

8. Klicken Sie auf **Anmeldenamen festlegen**. Daraufhin wird das Dialogfenster 'Anmeldenamen festlegen' geöffnet. Gehen Sie in diesem Dialogfenster wie folgt vor:

a. Geben Sie im Feld **Anmeldenamen hinzufügen** den Anmeldenamen für den TMR-Server eines Benutzers ein, der Zugriff auf UserLink erhalten soll.

b. Drücken Sie die Eingabetaste. Daraufhin wird der Name in die Liste **Aktuelle Anmeldenamen** übernommen.

c. Wiederholen Sie die Schritte 1 und 2 für alle Benutzer, die Zugriff auf UserLink erhalten sollen.

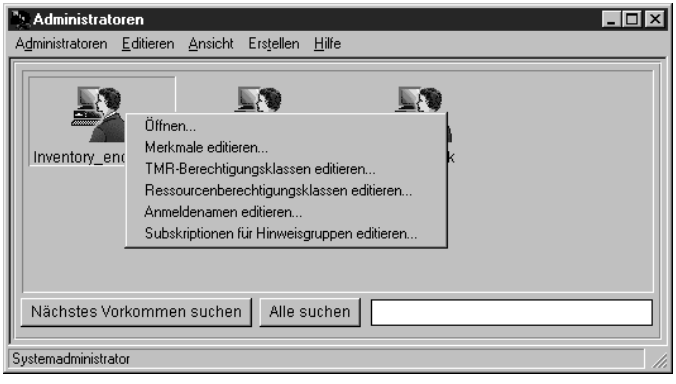
d. Klicken Sie anschließend auf **Festlegen und schließen**, um Ihre Änderungen zu sichern und in den Dialog 'Administrator erstellen' zurückzukehren

9. Klicken Sie auf **Erstellen und schließen**.

Daraufhin wird das Symbol des von Ihnen erstellten Administrators im Administratorfenster Ihrer TMR angezeigt.

Folgende Schritte sind zur Änderung der Administratoreinstellungen über die Arbeitsoberfläche erforderlich:

1. Klicken Sie mit der rechten Maustaste auf das Administratorsymbol, das geändert werden soll; daraufhin wird ein Kontextmenü aufgerufen.



2. Wählen Sie die folgenden Menüpunkte aus, um die Einstellungen für den Administratorzugriff auf UserLink zu ändern:

- **TMR-Berechtigungsklassen editieren** - Über diesen Menüpunkt können Sie dem Benutzer Administratorzugriff auf UserLink for Tivoli Inventory erteilen.
- **Anmeldenamen editieren** - Über diesen Menüpunkt können Sie die Anmeldenamen der Benutzer hinzufügen, ändern oder löschen, denen dieser Administrator zugewiesen ist und die auf UserLink zugreifen können.

Über die Befehlszeile

Mit dem Befehl **wcrtadmin** können Sie Benutzern Zugriff auf UserLink erteilen.

Mit dem folgenden Befehl wird der Administrator **end_user_admin** erstellt und diesem Zugriff auf UserLink erteilt; außerdem werden ihm die Benutzeranmeldenamen **jbrown**, **mgonzales** und **cwu** zugeordnet:

```
wcrtadmin –l jbrown,mgonzales,cwu \
–r @distinguished:InventoryUserLink,\
Inventory_end_user end_user_admin
```

Dabei gilt:

–l jbrown,mgonzales,cwu
: Gibt die Benutzer-IDs für den TMR-Server der Benutzer an, die als Administratoren eingerichtet werden sollen.

–r @distinguished:InventoryUserLink
: Gibt das spezifische Objekt an, das bei der Installation von Tivoli Inventory erstellt wird und das die Methoden für UserLink-Operationen enthält.

Inventory_end_user
: Gibt die Tivoli-Berechtigungsklasse an, die dem Administrator **end_user_admin** zugewiesen wird.

end_user_admin
: Gibt die Bezeichnung des Administratorsymbols auf der Tivoli-Arbeitsoberfläche an.

Sie können mit dem Befehl **wsetadmin** zu einem späteren Zeitpunkt weitere Anmeldenamen hinzufügen. Darüber hinaus haben Sie die

Möglichkeit, einem bereits vorhandenen Administrator mit dem Befehl **wsetadmin** Zugriff auf UserLink for Tivoli Inventory zu erteilen. Mit dem folgenden Befehl erhält der Administrator **end_user_admin** die Zugriffsberechtigung für Tivoli Inventory.

```
wsetadmin –r @distinguished:InventoryUserLink,Inventory_end_user \
end_user_admin
```

Dabei gilt:

–r @distinguished:InventoryUserLink
: Gibt das spezifische Objekt an, das bei der Installation von Tivoli Inventory erstellt wird und das die Methoden für UserLink-Operationen enthält.

Inventory_end_user
: Gibt die neue Berechtigungsklasse an, die dem Administrator zugewiesen werden soll.

end_user_admin
: Gibt den Administrator an, dessen Einstellungen geändert werden.

Weitere Informationen zur Syntax und Verwendung der Befehle **wcrtadmin** und **wsetadmin** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen. Neben der Eingabe einzelner neuer Anmeldenamen über die Tivoli-Arbeitsoberfläche oder den Befehl **wsetadmin** haben Sie auch die Möglichkeit, diesen Vorgang über ein eigens dafür erstelltes Skript automatisch auszuführen.

Tabellen für Benutzerdaten erstellen

Nachdem Sie festgelegt haben, welche Benutzerdaten erfasst werden sollen, müssen Sie entscheiden, wie diese im Konfigurations-Repository-Schema abgelegt werden. In den standardmäßig vorhandenen Tabellen des Datenbankschemas dürfen nur die Daten untergebracht werden, die bei einem Hardware- bzw. Softwaresuchlauf mit Tivoli Inventory erfasst werden. Wenn Sie mit Hilfe der Datei useradd.mif Benutzerdaten erfassen möchten, müssen Sie Tabellen und Spalten im Konfigurations-Repository zum Speichern von Daten erstellen.

Nachdem Sie die Daten festgelegt haben, die von den Benutzern abgerufen werden sollen, können Sie die Tabellen und Spalten erstellen, in denen diese Daten im Konfigurations-Repository gespeichert werden sollen. Dabei sollten Sie Folgendes berücksichtigen:

- Die Anzahl der neuen Tabellen sowie deren Anzahl Spalten. Beide hängen vom Umfang der Daten und von der Häufigkeit ab, mit der diese Daten erfasst werden.
- Die Bezeichnungen der neuen Tabellen und Spalten. Hierbei muss Folgendes berücksichtigt werden:
  - Die Tabellen- und Spaltenbezeichnungen müssen den Bezeichnungen der Gruppen und Attribute in der Datei 'useradd.mif' entsprechen. Wenn Sie beispielsweise eine neue Tabelle namens USER_INFO erstellen, muss der MIF-Gruppe ebenfalls die Bezeichnung USER_INFO zugewiesen werden. Die in der Gruppe USER_INFO in der Datei 'useradd.mif' enthaltenen Daten werden dann in der Tabelle USER_INFO gespeichert. Analog hierzu müssen Sie bei der Bezeichnung der Spalten in der Tabelle USER_INFO die Namen der MIF-Attribute in der MIF-Gruppe USER_INFO verwenden.
  - Spalten- und Tabellennamen für DB2- und Informix-Datenbanken dürfen nicht mehr als 18 Zeichen enthalten. Eine Liste mit Abkürzungen, wie sie für die Bezeichnung von Tabellen und Spalten in DB2-Datenbanken verwendet werden können, finden Sie im Abschnitt „Benutzerspezifische Tabellen, Anzeigen und Spalten für DB2 und Informix benennen" auf Seite 138. Wenn Sie eine Protokolltabelle für Ihre benutzerspezifische Tabelle erstellen möchten, dürfen Sie für den Namen der benutzerspezifischen Tabelle nicht mehr als 16 Zeichen eingeben. Weitere Informationen hierzu können Sie dem Abschnitt „Protokolltabellen für benutzerspezifische Tabellen erstellen" auf Seite 140 entnehmen.
  - Die Bezeichnungen für Tabellen und Spalten können folgende Zeichen enthalten:
    - Kleinbuchstaben von a bis z
    - Großbuchstaben von A bis Z

– Numerische Zeichen von 0 bis 9

– Unterstreichungen

Sie können keine Leerzeichen verwenden; Tabellen- und Spaltennamen müssen mit einem Buchstaben beginnen.

- Tabellen und Spalten für Benutzerdaten dürfen nicht dieselben Namen wie die im Konfigurations-Repository standardmäßig vorhandenen Tabellen bzw. Spalten erhalten. Die bereits vergebenen Tabellen- und Spaltenbezeichnungen können die dem Abschnitt „Konfigurations-Repository” auf Seite 365 oder dem Poster *Tivoli Inventory Configuration Repository* entnehmen.
- Sie dürfen keine SQL-92-reservierten Wörter für die Namen von benutzerspezifischen Tabellen und Spalten verwenden. Wenden Sie sich an den Datenbank-Administrator bzw. lesen Sie die SQL-Dokumentation, um weitere Informationen zu SQL-92-reservierten Wörtern zu erhalten.

- Legen Sie zunächst fest, welche Daten in den einzelnen Spalten gespeichert werden sollen. Die Spaltenarten müssen den Attributarten entsprechen, die Sie beim Hinzufügen von Attributen über die Benutzerdatenschablone auswählen.
- Bei der Anordnung der Tabellen im Konfigurations-Repository-Schema müssen Sie die Beziehung zwischen diesen Daten und anderen Tivoli Inventory-Daten berücksichtigen.
- Legen Sie fest, ob die neuen Tabellen zu bereits vorhandenen Tabellen im Schema in Beziehung gesetzt werden sollen. Sie können die von Ihnen hinzugefügten Tabellen bereits vorhandenen Tabellen über Primär- und Fremdschlüssel zuordnen. Dabei wird der Primärschlüssel einer im Konfigurations-Repository vorhandenen Tabelle als Fremdschlüssel in der neuen Tabelle definiert.
- In allen benutzerspezifischen Tabellen muss die Spalte COMPUTER_SYS_ID im Primärschlüssel enthalten sein.

Benutzerspezifische Tabellen, Anzeigen und Spalten für DB2 und Informix benennen

In DB2- und Informix-Datenbanken dürfen die Namen für Tabellen, Anzeigen und Spalten aus maximal 18 Zeichen bestehen. Wegen dieser Begrenzung kürzt das RDBMS Interface Module (RIM) die Tabellen-, Anzeigen- und Spaltennamen, die in der folgenden Tabelle beschrieben werden, ab. Verwenden Sie bei der Erstellung von benutzerspezifischen Tabellen, Anzeigen oder Spalten die Abkürzungen und nicht die vollständigen Namen.

Standardmäßiger Name	Abkürzung
ACTIVATION	ACTIV
ADDRESS	ADDR
ADMINISTRATOR	ADMIN
ARCHITECTURE	ARCH
AVAILABLE	AVAIL
BLOCK	BLK
BRIDGE	BRG
CHANGE	CHG
COMPONENT	COMP
CONFIG	CFG
CONNECTIONS	CONX
CONSOLE	CONS
CONTROLLER	CNTRL
CONVENTIONAL	CONV
COPROCESSOR	COPROC
DESCRIPTION	DESCRIP
DEVICE	DEV
DIRECTORY	DIR
DISTRIBUTION	DIST
DRIVERS	DRVS
DRIVER	DRV
FAXMODEM	FMODEM

Standardmäßiger Name	Abkürzung
FILEPACK	FPACK
FILESYSTEM	FILESYS
FLOPPYDRIVE	FLPY_DRV
HARDDISK	HDISK
HARDWARE	HWARE
HISTORY	HIST
INSTALLED	INST
INSTALLATION	INSTL
INTERNET	INET
INTERRUPT	INT
KERNEL	KRNL
LANGUAGE	LANG
LASTSCAN	LSCAN
LOGICALDRIVE	LOG_DRV
MEMORY	MEM
NETWARE	NWARE
NETWORK	NET
NUMBER	NUM
PLATFORM	PLTFRM
PROFILE	PRFL
RELATIONSHIP	REL
RESOLUTION	RES
RESTRICT	RSTRT
RWOPTICAL	RWOPT
SECURITY	SEC
SERIAL	SER
SERIALNUMBER	SERNUM
SERVICE	SVC
SIGNATURE	SIG
SOFTWARE	SW
START	ST

Standardmäßiger Name	Abkürzung
SUPPORT	SUPP
SYSTEM	SYS
SYSTEMS	SYS
VERSION	VER
VIDEO	VID
VIRTUAL	VIRT
VOLUME	VOL
VOLUMELABEL	VOLLABEL
DRIVE	DRV

Protokolltabellen für benutzerspezifische Tabellen erstellen

Um die Protokollfunktion für benutzerspezifische Tabellen zu verwenden, müssen Sie bei der Erstellung Ihrer benutzerspezifischen Tabelle eine Protokolltabelle erstellen. Der Name der Protokolltabelle muss mit dem Namen der benutzerspezifischen Tabelle übereinstimmen, wobei dem Namen der Protokolltabelle ein großgeschriebenes H und ein Unterstreichungszeichen (H_) vorangestellt werden muss. Beispiel: Wenn Ihre benutzerspezifische Tabelle die Bezeichnung MY_TABLE hat, muss der Name der Protokolltabelle für diese benutzerspezifische Tabelle folgendermaßen lauten: H_MY_TABLE.

Wenn Sie DB2- bzw. Informix-Datenbanken als Konfigurations-Repository verwenden, dürfen die Namen Ihrer Protokolltabellen gemäß der Namenskonvention für Protokolltabellen einschließlich des Präfixes H_ höchstens 18 Zeichen enthalten.

Die Spaltennamen der Protokolltabellen müssen mit den Spaltennamen der benutzerspezifischen Tabellen identisch sein. Der Datentyp einer Spalte in der Protokolltabelle muss mit dem Datentyp der entsprechenden Spalte in der benutzerspezifischen Tabelle übereinstimmen. Darüber hinaus muss die Protokolltabelle folgende Spalten enthalten:

Spaltenname	Datentyp
RECORD_ACTION	CHAR(6)
PRFL_ACTION	VARCHAR(20)
RECORD_TIME	ZEITMARKE oder für Ihr Verwaltungssystem für relationale Datenbanken geeignete Datenstruktur

Für Protokolltabellen gibt es in Tivoli Inventory keine Vorgaben. Wenn Sie eine benutzerspezifische Protokolltabelle ordnungsgemäß erstellen, wird diese von Tivoli Inventory automatisch erkannt und mit Daten ausgefüllt.

Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen

Für die Erfassung von Benutzerdaten mit Tivoli Inventory müssen Sie zunächst dem Konfigurations-Repository die neuen Tabellen und Spalten hinzufügen. Dabei müssen Sie den Tabellennamen, die Anzahl Spalten in der Tabelle, die Spaltenbreite sowie den Datentyp, der in den einzelnen Spalten gespeichert werden sollen, angeben und außerdem festlegen, ob die Spalten als Primär- und/oder Fremdschlüssel definiert werden sollen. Voraussetzung für die Erstellung neuer Tabellen sind Kenntnisse in Bezug auf SQL und die Grundlagen relationaler Datenbanken (z. B. Primärschlüssel, Fremdschlüssel, Datentypen, Nullfelder usw.).

Sie müssen beim Hinzufügen von benutzerspezifischen Tabellen auf Fehlernachrichten achten, die anzeigen, dass die Tabelle nicht ordnungsgemäß erstellt wurde.

Mit dem folgenden SQL-Beispiel wird die Tabelle USER_INFO dem Konfigurations-Repository in einer Oracle-Datenbank hinzugefügt. Diese Tabelle wird bei Änderungen aktualisiert und enthält die folgenden Spalten:

- LAST_NAME
- FIRST_NAME
- EMP_NUMBER

- OFFICE_NUMBER
- COMPUTER_SYS_ID

Die von den Benutzern im Benutzerdatenformular eingegebenen Daten (Name, Personalnummer und Raumnummer) werden im Konfigurations-Repository der entsprechenden Maschinen-ID (COMPUTER_SYS_ID) zugeordnet.

Die entsprechend dieser Tabelle erstellte Datei useradd.mif enthält die Gruppe USER_INFO, die die Attribute LAST_NAME, FIRST_NAME, EMP_NUMBER und OFFICE_NUMBER umfasst.

Anmerkung: Im Folgenden handelt es sich um ein Beispiel für ein Oracle-Konfigurations-Repository.

Um die Tabelle USER_INFO hinzuzufügen, müssen Sie sich zunächst im Konfigurations-Repository als Benutzer oder Eigner anmelden. Geben Sie anschließend in der Befehlszeile den folgenden Befehl ein:

```
DROP TABLE USER_INFO;

CREATE TABLE USER_INFO(
   COMPUTER_SYS_ID      varchar2(64) NOT NULL,
   LAST_NAME            varchar2(64) NOT NULL,
   FIRST_NAME           varchar2(64) NOT NULL,
   EMP_NUMBER           varchar2(32) NOT NULL,
   OFFICE_NUMBER        varchar2(12) NOT NULL,
Vorgabe USERINFO_PK PRIMARY KEY (COMPUTER_SYS_ID),
Vorgabe USERINFO_FK FOREIGN KEY (COMPUTER_SYS_ID)
Vorgabe
on delete cascade
);
commit;
```

Die Integritätsbedingung über Fremdschlüssel in diesem Skript ist erforderlich, wenn Sie unter Verwendung des Befehls **winvrmnode** alle Informationen zu einem Endpunkt aus dieser Tabelle entfernen möchten, wenn dieser Endpunkt gelöscht wird. Dank dieser Bedingung können Sie eine *Mehrstufige Löschoperation* ausführen: wenn Sie einen Wert aus der Tabelle COMPUTER löschen, auf den in dieser Tabelle durch einen Fremdschlüssel verwiesen wird, wird auch der Wert, der dem Fremdschlüssel in dieser Tabelle zugeordnet ist,

gelöscht. Wenden Sie sich an einen qualifizierten Datenbankadministrator, wenn Sie weitere Informationen zu mehrstufigen Löschoperationen haben möchten. Beispiele für mehrstufige Löschoperationen und den zugehörigen Auslösern, die für Ihr Verwaltungssystem für relationale Datenbanken geeignet sind, können Sie den Installationsskripts für Schemata im Verzeichnisskript in $BINDIR/../generic/inv/SCRIPTS/RDBMS entnehmen.

Geben Sie den folgenden Befehl ein, wenn Sie eine Protokolltabelle für die Tabelle USER_INFO erstellen möchten:

```
DROP TABLE H_USER_INFO;

CREATE TABLE H_USER_INFO(
   COMPUTER_SYS_ID      varchar2(64),
   LAST_NAME            varchar2(64),
   FIRST_NAME           varchar2(64),
   EMP_NUMBER           varchar2(32),
   OFFICE_NUMBER        varchar2(12),
   RECORD_ACTION        char(6),
   PRFL_ACTION          varchar2(20),
   RECORD_TIME          DATE
);
commit;
```

Beachten Sie, dass es für die Protokolltabelle keine Vorgaben gibt.

Zum Abrufen von Benutzerdaten können Sie außerdem eine neue Anzeige erstellen. Weitere Informationen hierzu können Sie dem Abschnitt „Benutzerdaten anzeigen” auf Seite 156 entnehmen. Sie können auch mit dem Befehl **wcrtquery** Ihrer Abfragebibliothek benutzerspezifische Abfragen hinzufügen. Weitere Informationen können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

MIF-Gruppen und -Attribute anhand der Benutzerdatenschablone erstellen

Nachdem Sie dem Konfigurations-Repository ein oder mehrere Tabellen zur Speicherung von Benutzerdaten hinzugefügt haben, müssen Sie nun in einem nächsten Schritt über die Benutzerdatenschablone die entsprechenden MIF-Gruppen und -Attribute erstellen; bei der Schablone handelt es sich um eine Gruppe von HTML-Formularen, die Ihnen in Tivoli Inventory zur Verfügung stehen.

Dabei werden als Tabellen- bzw. Spaltenbezeichnungen die Namen der in der Benutzerdatenschablone angegebenen MIF-Gruppen und -Attribute verwendet. Für jede neue Tabelle, die Sie dem Konfigurations-Repository hinzufügen, müssen Sie eine Gruppe erstellen, der als Name die Tabellenbezeichnung zugewiesen wird. Der Gruppenname muss der Tabellenbezeichnung exakt entsprechen, einschließlich Groß-/Kleinschreibung. Für die Tabelle BENUTZER-INFO muss demzufolge eine MIF-Gruppe mit dem Namen BENUTZERINFO erstellt werden. Für jede Spalte in der Tabelle müssen Sie ein entsprechendes Attribut in der MIF-Gruppe erstellen. So umfasst beispielsweise die Gruppe BENUTZERINFO die Attribute NACHNAME, VORNAME, PERSONALNR und TEL_GESCHAEFT, für die in der Tabelle BENUTZERINFO entsprechende Spalten mit den identischen Bezeichnungen vorliegen.

Sollen andere benutzerspezifische Informationen erfasst werden, können Sie die aktuellen MIF-Gruppen bzw. -Attribute in der Benutzerdatenschablone löschen. Soll eine MIF-Gruppe entfernt werden, müssen Sie zunächst alle Attribute der betreffenden Gruppe löschen. Nach dem Entfernen eines MIF-Attributs wird das entsprechende Eingabefeld im Benutzerdatenformular beim nächsten Zugriff nicht mehr angezeigt und die gleichnamige Spalte im Konfigurations-Repository nicht mehr aktualisiert. Sie haben außerdem die Möglichkeit, weitere Attribute hinzuzufügen, um zusätzliche Daten zu erfassen. Dazu müssen Sie jedoch zunächst den Tabellen im Konfigurations-Repository die entsprechenden Spalten hinzufügen.

Folgende Schritte sind zur Erstellung der Gruppe BENUTZERINFO und deren Attribute über die administratorspezifischen Web-Seiten von Tivoli Inventory erforderlich:

1. Rufen Sie die folgende URL in einem Web-Browser auf, um zur Web-Seite mit der Benutzerdatenschablone zu gelangen:

   http://*Server:Oserv-Port-ID*

   Dabei steht *Server* für den Host-Namen des TMR-Servers und *Oserv-Port-ID* für den Anschluss, an dem der Prozess oserv empfangsbereit ist. Standardmäßig ist dem Prozess oserv die Anschlussnummer 94 zugeordnet. Beispiel: **http://ball:94**.

   Die Seite ’TME 10 Web Access’ wird angezeigt.

2. Klicken Sie auf den Link **Tivoli Inventory**. Daraufhin wird im Web-Browser ein Dialog mit Eingabefeldern für den Benutzernamen und das Kennwort aufgerufen. (Falls der Link 'Tivoli Inventory' nicht vorhanden ist, ist Tivoli Inventory noch nicht installiert.)

3. Geben Sie Ihren Tivoli-Administratoranmeldenamen und das Kennwort ein, mit dem Sie sich an der Maschine angemeldet haben. Überprüfen Sie im Falle einer Zugriffsverweigerung folgende Einstellungen:

   - Für die Ausführung des Befehls **odadmin** muss die Option "Remote client login allowed" auf TRUE gesetzt sein. Ist dies nicht der Fall, geben Sie den folgenden Befehl ein:

     ```
     odadmin set_allow_rconnect TRUE
     ```

   - Sie müssen über die Administratorberechtigungsklassen **RIM_update** und **RIM_view** verfügen.

   - Auf einem TMR-Server, der unter UNIX läuft, muss die Benutzer-ID und das Kennwort als zulässige Anmeldekennung in der Datei **/etc/passwd** enthalten sein.

   - Wenn es sich bei dem TMR-Server um einen NT-Server handelt, der als primärer (PDC) bzw. Sicherungs-Domänen-Controller (SDC) fungiert, müssen Sie *Domäne\Anmeldename* im Feld **Benutzername** eingeben; dabei steht *Domäne* für die Domäne die von dem primären bzw. dem Sicherungs-Domänen-Controller gesteuert wird, und *Anmeldename* für den Anmeldenamen des Administrators.

     Anmerkung: Nach einer erfolgreichen Authentifizierung haben Sie Zugriff auf die Tivoli Inventory-Web-Seiten, solange der Browser aktiv ist.

4. Klicken Sie auf den Link **Benutzerdatenschablone**. Die Benutzerdatenschablone von Tivoli Inventory wird geöffnet.

5. Folgende Schritte sind zum Hinzufügen der MIF-Gruppe USER-_INFO erforderlich:

   a. Klicken Sie auf den Link **MIF-Gruppe hinzufügen**. Daraufhin wird die Seite 'MIF-Gruppen hinzufügen' angezeigt.

MIF-Gruppe hinzufügen
Aktuelle MIF-Gruppen
MIF-Gruppe, die hinzugefügt werden soll
Geben Sie einen Namen und eine Beschreibung für die MIF-Gruppe ein, die hinzugefügt werden soll.
Der Gruppenname wird im Konfigurations-Repository als Tabellenname verwendet. Geben Sie den Namen in Großbuchstaben und ohne Leerzeichen ein.
Gruppenname
Gruppenbeschreibung
Hinzufügen
Zurücksetzen
Zurück zur Benutzerdatenschablone

b. Geben Sie im Feld ’Gruppenname’ den Namen einer MIF-Gruppe wie beispielsweise **BENUTZERINFO** ein. Geben Sie den Namen in Großbuchstaben ein. Bitte verwenden Sie keine Leerzeichen, da es sich bei diesem Namen auch um einen Tabellennamen im Konfigurations-Repository handelt. Er darf daher keine Leerstellen enthalten.

c. Geben Sie im Feld ’Gruppenbeschreibung’ eine Beschreibung der MIF-Gruppe ein, z.B. **Informationen zum Personal**. Die Angabe einer Beschreibung ist für DMI Version 2.0 erforderlich (Desktop Management Interface); diese Beschreibung wird im Benutzerdatenformular angezeigt.

d. Klicken Sie auf **Hinzufügen**, um die MIF-Gruppe hinzuzufügen. Daraufhin wird eine Seite mit dem Ergebnis des Vorgangs geöffnet.

Ergebnis

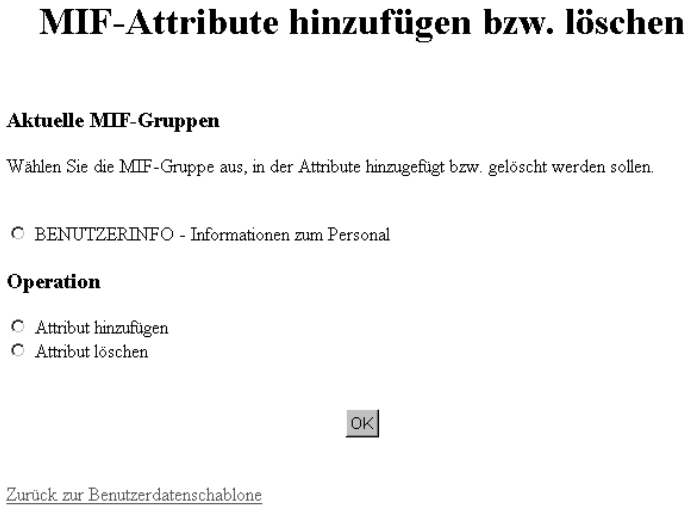
MIF-Gruppe "BENUTZERINFO" wurde hinzugefügt.

Zurück zur Benutzerdatenschablone

6. Soll der Gruppe BENUTZERINFO ein Attribut hinzugefügt werden, klicken Sie zunächst auf den Link **Zurück zur Benutzerdatenschablone**. Daraufhin wird die Benutzerdatenschablone wieder geöffnet.

   a. Klicken Sie auf den Link **MIF-Attribute hinzufügen bzw. löschen**.

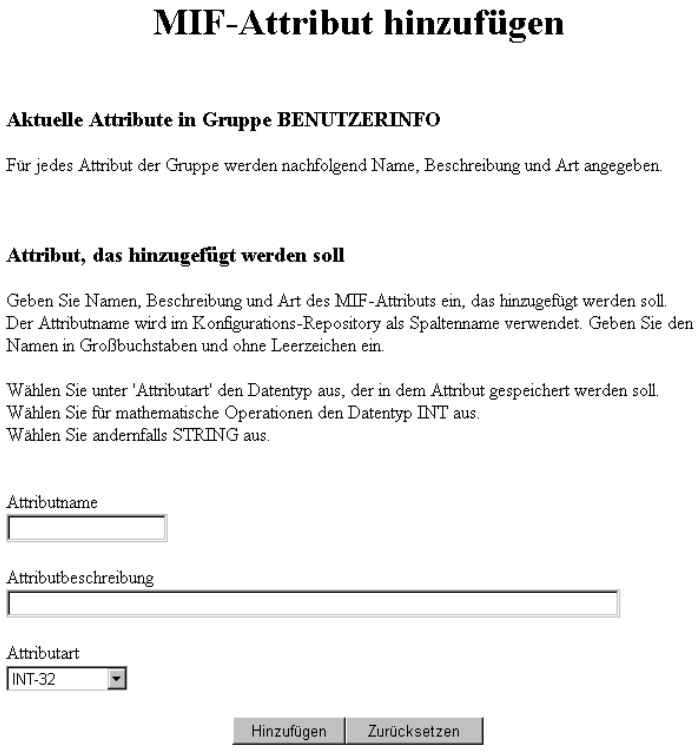
Daraufhin wird die Seite ’MIF-Attribute hinzufügen bzw. löschen’ angezeigt.



b. Wählen Sie die MIF-Gruppe aus, der ein Attribut hinzugefügt werden soll.

c. Wählen Sie die Option **Attribut hinzufügen** aus, und klicken Sie auf **OK**.

Daraufhin wird die Seite ’MIF-Attribut hinzufügen’ angezeigt.


MIF-Attribut hinzufügen
Aktuelle Attribute in Gruppe BENUTZERINFO
Für jedes Attribut der Gruppe werden nachfolgend Name, Beschreibung und Art angegeben.
Attribut, das hinzugefügt werden soll
Geben Sie Namen, Beschreibung und Art des MIF-Attributs ein, das hinzugefügt werden soll.
Der Attributname wird im Konfigurations-Repository als Spaltenname verwendet. Geben Sie den Namen in Großbuchstaben und ohne Leerzeichen ein.
Wählen Sie unter 'Attributart' den Datentyp aus, der in dem Attribut gespeichert werden soll.
Wählen Sie für mathematische Operationen den Datentyp INT aus.
Wählen Sie andernfalls STRING aus.
Attributname
Attributbeschreibung
Attributart
INT-32
Hinzufügen
Zurücksetzen

d. Geben Sie im Feld ’Attributname’ einen Namen für das Attribut ein, z. B. **NACHNAME**. Bitte beachten Sie, dass es sich bei dieser Angabe auch um den Namen einer Spalte im Konfigurations-Repository handelt; der Name darf daher keine Leerstellen enthalten, da diese in Tabellen bzw. Spalten des Konfigurations-Repositorys nicht zulässig sind.

e. Geben Sie im Feld ’Attributbeschreibung’ eine Beschreibung für das Attribut ein, z.B. **Nachname**. Diese Beschreibung wird im Benutzerdatenformular neben dem entsprechenden Textfeld angezeigt.

f. Wählen Sie in der Liste ’Attributart’ einen Eintrag aus. Sollen die Daten für mathematische Berechnungen verwendet werden, wählen Sie **INT** aus. Wählen Sie andernfalls **STRING** aus. Der hier ausgewählte Datentyp muss dem Datentyp entsprechenden, den Sie für die entsprechende Spalte angegeben haben. Wählen Sie für das vorliegende Beispiel den Eintrag **STRING-64** aus, da dieser dem Datentyp sowie der Breite der Spalte NACHNAME in der Tabelle BENUTZERINFO entspricht, die wie unter „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen” auf Seite 141 beschrieben hinzugefügt wurde.

g. Klicken Sie auf **Hinzufügen**, um das Attribut der MIF-Gruppe hinzuzufügen. Eine Seite mit den Ergebnissen des Vorgangs wird angezeigt.

Ergebnis

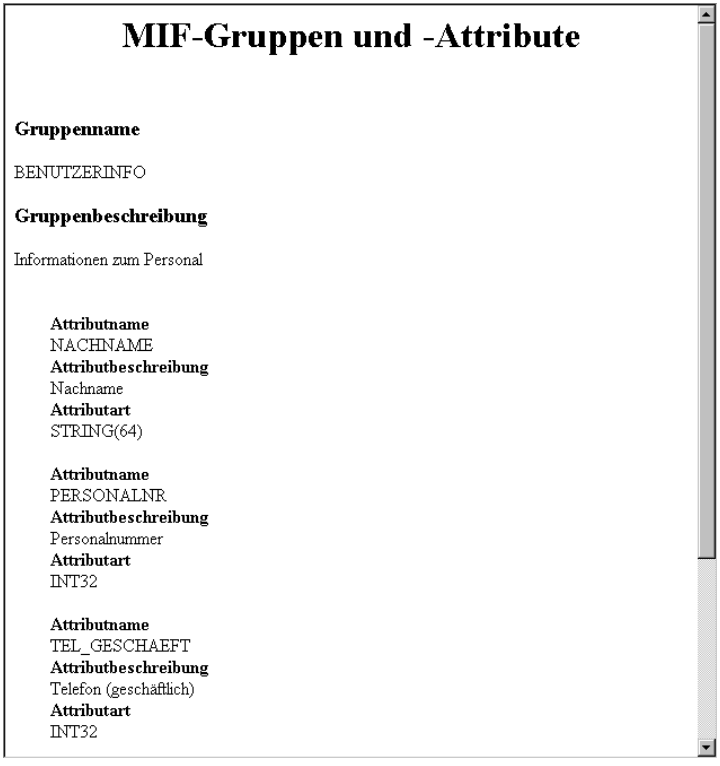
Das MIF-Attribut "NACHNAME" wurde hinzugefügt.

Hinzufügen bzw. Löschen eines weiteren MIF-Attributs

Zurück zur Benutzerdatenschablone

h. Wiederholen Sie die Schritte 6a bis 6g für alle MIF-Gruppen und -Attribute, die der Datei useradd.mif hinzugefügt werden sollen.

7. Klicken Sie auf auf den Link **MIF-Gruppen und -Attribute anzeigen**, um eine Liste der aktuellen MIF-Gruppen und -Attribute anzuzeigen. Daraufhin wird die Seite ’MIF-Gruppen und -Attribute’ angezeigt.



MIF-Gruppen und -Attribute

Gruppenname

BENUTZERINFO

Gruppenbeschreibung

Informationen zum Personal

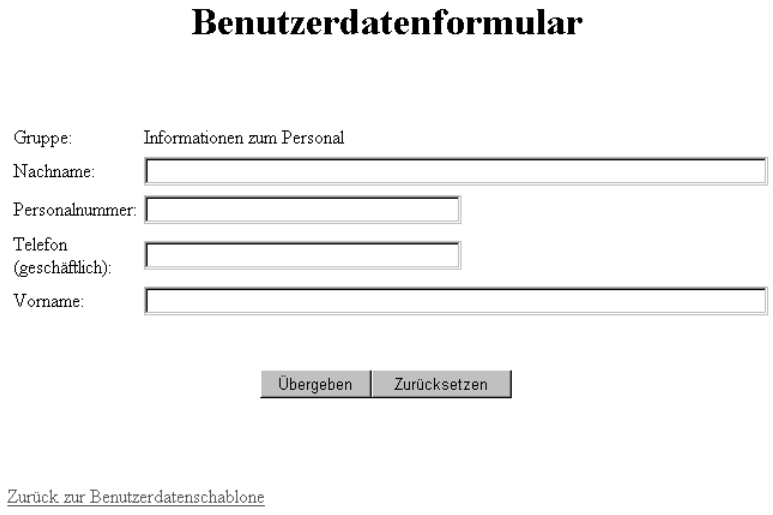
Attributname
NACHNAME
Attributbeschreibung
Nachname
Attributart
STRING(64)

Attributname
PERSONALNR
Attributbeschreibung
Personalnummer
Attributart
INT32

Attributname
TEL_GESCHAEFT
Attributbeschreibung
Telefon (geschäftlich)
Attributart
INT32

8. Klicken Sie auf der Seite ’Benutzerdatenschablone’ auf den Link ’Voranzeige Benutzerdatenformular’, um die grafische Umsetzung der von Ihnen eingegebenen Informationen im Benutzerdatenformular zu überprüfen.

Daraufhin wird das Benutzerdatenformular so angezeigt, wie es sich den Benutzern darstellt.



Datei userlink.htm verwenden

Um eine Verbindung zu den Funktionen von UserLink for Tivoli Inventory herzustellen, müssen die Benutzer die HTML-Datei userlink.htm in einem Web-Browser öffnen. Endpunkte erhalten die Datei userlink.htm bei der Anmeldung am Gateway. Die Datei ist im Verzeichnis $DRIVE:/etc/Tivoli/$LANG gespeichert; dabei steht $DRIVE für das Laufwerk, auf dem der Tivoli-Agent installiert ist, und $LANG für die auf dem Endpunkt verwendete Sprache. Weitere Informationen zu $LANG können Sie den *Tivoli Management Framework Release-Notes* entnehmen.

Anmerkung: Benutzern wird empfohlen, die Seite userlink.htm der Liste mit den Lesezeichen in ihrem Web-Browser hinzuzufügen; dadurch wird der Zugriff auf das Benutzerdatenformular vereinfacht.

Benutzerdatenformular ausfüllen

Weisen Sie die Benutzer an, UserLink for Tivoli Inventory aufzurufen und die gewünschten Informationen im Benutzerdatenformular einzugeben. Dieses Formular wird bei jedem Zugriff auf den Link 'Benutzerdatenformular' über UserLink erstellt. Weisen Sie die Benutzer an, vor jeder Dateneingabe das Benutzerdatenformular neu zu laden; dadurch wird sichergestellt, dass das Formular immer die aktuellen Eingabefelder enthält.

Im Anschluss an die Dateneingabe im Benutzerdatenformular wird die Datei useradd.mif auf dem Endpunkt im Verzeichnis $LCF_BASE_DIR/inv/SCAN erstellt. Dabei steht $LCF_BASE_DIR für das Verzeichnis, in dem der Tivoli-Agent installiert ist, und $oid für die Tivoli-Objekt-ID des Clients.

Es wird empfohlen, die Datei useradd.mif entweder sofort abzurufen und im Konfigurations-Repository zu speichern oder nach der Erstellung in ein anderes Verzeichnis zu verschieben: Die Datei useradd.mif wird jedesmal überschrieben, wenn ein Benutzer im Benutzerdatenformular auf **Übergeben** klickt.

Folgende Schritte sind erforderlich, um UserLink aufzurufen und das Benutzerdatenformular auszufüllen:

1. Öffnen Sie in einem Web-Browser die Datei userlink.htm. Informationen zum Pfad können Sie dem Abschnitt „Datei userlink.htm verwenden" auf Seite 152 entnehmen.

   Die allgemeine Seite von UserLink for Tivoli wird geöffnet.

2. Klicken Sie auf den Link **UserLink for Tivoli**. Daraufhin wird die Web-Seite von UserLink for Tivoli aufgerufen.

UserLink for Tivoli

UserLink for Tivoli ermöglicht den Fernzugriff auf folgende Tivoli-Anwendungsschnittstellen. Wählen Sie eine der folgenden Verknüpfungen, um eine Verbindung zu einer UserLink for Tivoli-Anwendungsschnittstelle herzustellen. Beachten Sie bei Verwendung dieser Funktionen die Anweisungen Ihres Systemadministrators.

UserLink for Tivoli Inventory

Füllen Sie auf Anforderung des Administrators ein Benutzerdatenformular mit Hilfe von UserLink for Tivoli Inventory aus. In diesem Formular werden die Daten gesammelt, die bei einem Inventory-Suchlauf nicht automatisch gesammelt werden.

3. Klicken Sie auf den Link **UserLink for Tivoli Inventory**. Daraufhin wird die Seite von UserLink for Tivoli Inventory aufgerufen.

UserLink for Tivoli Inventory

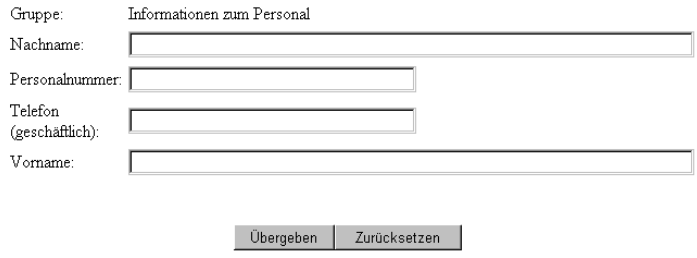
Willkommen bei UserLink for Tivoli Inventory. Bei Inventory handelt es sich um ein Softwareprogramm für Administratoren zum Erfassen von Software- und Hardwaredaten in einer Network Computing-Umgebung. Benutzer unterstützen die Erfassung von Hard- und Softwareinformationen mit Hilfe der folgenden Inventory-Funktionen.
Diese Funktionen sollten nur entsprechend den Anweisungen des System- oder Netzwerkadministrators verwendet werden.

Benutzerdatenformular

Diese Web-Seite enthält das Benutzerdatenformular. In diesem Formular können Sie Daten eingeben, die bei Hardware- oder Softwaresuchläufen von Inventory nicht ermittelt werden.

Mon Mar 5 14:57:29 2001

4. Klicken Sie auf den Link **Benutzerdatenformular**. Daraufhin wird ein Dialogfenster geöffnet, in dem Sie zur Eingabe des Benutzernamens und des Kennworts aufgefordert werden.
5. Geben Sie Namen und Kennwort ein, die für den Endbenutzeradministrator eingerichtet wurde. Daraufhin wird das Benutzerdatenformular angezeigt.



6. Geben Sie in den Feldern des Benutzerdatenformulars die entsprechenden Informationen ein.
7. Klicken Sie auf **Übergeben**. (Im Anschluss an die Dateneingabe im Benutzerdatenformular wird die Datei useradd.mif lokal erstellt.)

Benutzerdaten abrufen und speichern

Nachdem die Benutzer das Benutzerdatenformular an ihren Maschinen ausgefüllt haben, können Sie die Datei useradd.mif von den Maschinen der Benutzer abrufen und die Daten im Konfigurations-Repository speichern. Bei der Datei useradd.mif handelt es sich um die MIF-Datei mit den Daten, die vom Benutzer eingegeben wurden. Bei der Verteilung werden die Daten in der Datei useradd.mif in den Tabellen im Konfigurations-Repository gespeichert, die Sie für diesen Zweck erstellt haben. Verteilen Sie ein Inventory-Profil, das zum Lesen der Datei useradd.mif angepasst wurde, um die Datei useradd.mif abzurufen. Um das Inventory-Profil anzupassen, müssen Sie useradd.mif einschließlich des Pfads im Feld für benutzerspezifische MIF-Dateien im Fenster 'Skripts und MIF-Dateien' eingeben.

Benutzerdaten anzeigen

RDBMS-Anzeigen werden erstellt, um den Zugriff auf Datenkategorien bei Abfragen zu erleichtern. Die Erstellung von Anzeigen ermöglicht die Anordnung von Daten, die sich in einander zugeordneten Tabellen befinden. So befinden sich beispielsweise die Spalten PROCESSOR_ MODEL, PROCESSOR_SPEED und COMPUTER-_SYS_ID in verschiedenen Tabellen des Konfigurations-Repositorys. Die Anzeige PROCESSOR_VIEW enthält alle Daten, die in diesen Spalten enthalten sind. Dadurch können Sie alle gewünschten Informationen über die Anzeige PROCESSOR_VIEW abrufen, anstatt für jede Tabelle eine Abfrage zu starten.

Die in Tivoli Inventory zur Verfügung gestellten Anzeigen enthalten keine benutzerspezifischen Tabellen oder Spalten, die dem Konfigurations-Repository hinzugefügt wurden. Für die Abfrage von Benutzerdaten müssen Sie eine neue Anzeige erstellen. Hierzu erstellen Sie ein Skript, mit der diese neue Anzeige dem Konfigurations-Repository hinzugefügt wird.

Folgendes sollten Sie bei der Erstellung neuer Anzeigen berücksichtigen:

- Die standardmäßig in Tivoli Inventory zur Verfügung gestellten Anzeigen dürfen nicht geändert werden.

- Stellen Sie sicher, dass die neue Anzeige die Spalten TME_OBJECT_ID und TME_OBJECT_LABEL enthält. Wenn Sie unter Verwendung des Befehls **wruninvquery** eine Subskriptionsliste erstellen, führen Sie eine der in Tivoli Inventory verfügbaren Subskriptionsabfragen aus, oder führen Sie eine neue Abfrage aus, bei der die Namen der Maschinen zurückgegeben werden, die den Suchkriterien entsprechen.
- Fügen Sie der Tabelle QUERY_VIEWS im Konfigurations-Repository die neue Anzeige hinzu. Andernfalls wird die Anzeige beim Klicken auf die Schaltfläche mit den Auslassungszeichen (**...**) im Dialog 'Abfrage erstellen' bzw. 'Abfrage editieren' nicht angezeigt. Weitere Informationen zur Auswahl von Anzeigen im Abfragekatalog können Sie dem Abschnitt „Abfragen erstellen" auf Seite 211 entnehmen.

Mit dem folgenden Beispiel wird die Anzeige LOCATION_USER_VIEW in einem Konfigurations-Repository auf einem Oracle-Server erstellt. Über diese Anzeige können Sie die Daten aus den Spalten LAST_NAME, FIRST_NAME, EMP_NUMBER und OFFICE_NUMBER der Tabelle USER_INFO abfragen, die im Abschnitt „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen" auf Seite 141 erstellt wurde. Beachten Sie, dass einige Zeilen in der **SELECT-Anweisung** aus einem Tabellennamen gefolgt von einem Spaltennamen bestehen; die beiden Angaben sind jeweils durch einen Punkt getrennt. Bei Angabe einer Spalte, die in mehreren Tabellen enthalten ist, müssen Sie in der **SELECT-Anweisung** zuerst den Tabellennamen gefolgt von einem Punkt und dann den Tabellennamen eingeben.

Anmerkung: Mit dem folgenden Beispiel wird eine Anzeige in einer Oracle-Datenbank erstellt.

```
drop view LOCATION_USER_VIEW
create view LOCATION_USER_VIEW
as
select
TME_OBJECT_ID,
TME_OBJECT_LABEL,
LAST_NAME,
FIRST_NAME,
EMP_NUMBER,
```

```
OFFICE_NUMBER,
USER_INFO.COMPUTER_SYS_ID
from USER_INFO, COMPUTER
where
        USER_INFO.COMPUTER_SYS_ID =
             COMPUTER.COMPUTER_SYS_ID;

commit;
insert into QUERY_VIEWS values ('LOCATION_USER_VIEW');
commit;
```

Speichern Sie die Datei nach ihrer Erstellung unter *Dateiname***.sql**, und schließen Sie sie. Anschließend können Sie dem Konfigurations-Repository die neue Anzeige hinzufügen, indem Sie als Benutzer 'invtiv' diese Datei mit einem interaktiven SQL-Tool ausführen.

Sie können die neue Anzeige in bereits vorhandenen oder neu erstellten Abfragen verwenden, indem Sie entweder den Namen der Anzeige im Feld **Tabellen-/Anzeigenname** in den Dialogfenstern 'Abfrage erstellen', 'Abfrage editieren' und 'Inventory-Abfrage' eingeben oder auf die Schaltfläche mit den Auslassungszeichen (**...**) neben dem Feld **Tabellen-/Anzeigenname** klicken (vorausgesetzt der Name der Anzeige wurde als Zeile in der Tabelle **QUERY_VIEWS** hinzugefügt).

MIF-Benutzerdateien verwenden

Bei der Verteilung von Inventory-Profilen können alle MIF-Dateien gelesen werden, die dem MIF-Format des DMTF-Standards 2.0 (Desktop Management Task Force) entsprechen. Dadurch können Sie mit Tivoli Inventory auch Daten abfragen, die nicht bei einem Software- oder Hardwaresuchlauf erstellt wurden. Sie können MIF-Benutzerdateien abrufen, indem Sie für Inventory-Profile Einstellungen vornehmen, mit denen bei einer Profilverteilung MIF-Dateien gelesen werden können. Bei einer MIF-Benutzerdatei kann es sich um MIF-Dateien handeln, die mit einem anderen Programm oder manuell von Ihnen selbst erstellt wurden. Von Tivoli Inventory werden MIF-Software- bzw. MIF-Hardwaredateien bei Software- bzw. Hardwaresuchläufen erstellt.

Der Inhalt dieser MIF-Dateien wird in den standardmäßig vorhandenen Tabellen im Konfigurations-Repository gespeichert. Sollen MIF-Benutzerdateien verwendet werden, müssen Sie die entsprechenden Tabellen und Spalten im Konfigurations-Repository erstellen, in denen die Benutzerdaten gespeichert werden.

Um sicherzustellen, dass die MIF-Benutzerdateien von Tivoli Inventory gelesen werden können, müssen Sie Folgendes berücksichtigen:

- Die MIF-Datei muss den MIF-Normen des DMTF-Standards 2.0 entsprechen; unter anderem müssen folgende Bedingungen erfüllt sein:
  - Jeder Gruppe bzw. jedem Attribut muss ein Name und eine Beschreibung zugewiesen werden.
  - Für jedes Attribut müssen Art und Größe angegeben werden.
- Beachten Sie die Richtlinien zur Erstellung benutzerspezifischer Tabellen in den Abschnitten „Tabellen für Benutzerdaten erstellen" auf Seite 135 und „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen" auf Seite 141.
- Die standardmäßig im Konfigurations-Repository vorhandenen Tabellen dürfen nicht geändert werden.
- MIF-Gruppen dürfen nicht dieselben Namen wie die im Konfigurations-Repository standardmäßig vorhandenen Tabellen erhalten.
- Wenn Sie in der MIF-Datei mehrere Einträge innerhalb einer Gruppe in Tabellen speichern, müssen Sie das eindeutige Attribut bzw. die eindeutigen Attribute als Primärschlüssel verwenden. Diese Attribute müssen anschließend als Teil des Primärschlüssels in der Tabelle angegeben werden, die Sie für die MIF-Gruppe erstellt haben.

- Stellen Sie sicher, dass der Name der MIF-Benutzerdatei innerhalb der TMR eindeutig ist. Namen von MIF-Dateien, die von Tivoli Inventory bei Suchläufen erstellt werden, dürfen beispielsweise nicht verwendet werden. Von Tivoli Inventory werden die folgenden MIF-Dateien generiert:
  - dmiscan.mif
  - mrmbios.mif
  - tivfscan.mif
  - tivhscan.mif
  - tivrscan.mif
  - tivsscan.mif
  - tivwscan.mif
  - useradd.mif

Mit dem folgenden Beispiel werden einer MIF-Datei eine neue Gruppe und ein Attribut hinzugefügt.

```
START COMPONENT
NAME = "CUSTOMDATA.MIF"
DESCRIPTION = "Benutzerdaten, die bei einem Suchlauf
nicht erfasst werden"
   START GROUP
      NAME = "SYS_INFO"
         ID = 1
      START ATTRIBUTE
         NAME = "SERIAL_NUMBER"
         ID = 1
         ACCESS = READ-ONLY
         TYPE = STRING(32)
         VALUE = "123456789012"
      END ATTRIBUTE
      START ATTRIBUTE
         NAME = "MAKER"
         ID = 2
         ACCESS = READ-ONLY
         TYPE = STRING(32)
         VALUE = "Compaq"
      END ATTRIBUTE
   END GROUP
END COMPONENT
```

PC-Suchläufe für die Erfassung von BIOS-Daten anpassen

Sie können Tivoli Inventory konfigurieren, um PC-Systeme nach BIOS-Daten (Basic Input/Output System) zu durchsuchen, unter anderem nach dem Datum des BIOS-Chips, den BIOS-Kennbytes, der eindeutigen Zeichenfolge, dem Namen des Herstellers sowie dem Modell der Maschine, die durchsucht wird. Diese Informationen werden im Konfigurations-Repository von Tivoli Inventory in der Tabelle PC_SYS_PARAMS gespeichert.

Zur BIOS-Suchsoftware gehört ein Verzeichnis mit einer Reihe vorgegebener Maschinenhersteller und -modelle. Informationen zu Herstellern bzw. Modellen, die nicht im Verzeichnis enthalten sind, können der Datei bios.ini mit Hilfe des Dienstprogramms mrmbios hinzugefügt werden.

Folgende Schritte sind zum Aktualisieren der Datei bios.ini erforderlich:

1. Durchsuchen Sie alle PCs in Ihrer Umgebung nach BIOS-Daten, falls Sie dies nicht bereits getan haben.
2. Ermitteln Sie zunächst die PCs im Netz, deren Hersteller bzw. Modell unbekannt ist; gehen Sie dazu wie folgt vor:
   a. Fragen Sie die Tabelle PC_SYS_PARAMS oder die Anzeige PC_BIOS_VIEW ab.
   b. Überprüfen Sie anhand der Spalte BIOS_MODEL, ob das Modell einer Maschine unbekannt ist. Ist dies der Fall, enthält diese Spalte den Wert UNKNOWN.
   c. Ermitteln Sie für alle unbekannten Modellnummern die Spalte COMPUTER_SYS_ID, um festzustellen, von welchen Systemen der Hersteller bzw. das Modell nicht bekannt ist.

3. Führen Sie zum Aktualisieren der Datei bios.ini das Dienstprogramm mrmbios über die Befehlszeile auf jedem unter Schritt 2 ermittelten PC aus. Geben Sie den folgenden Befehl ein:

   ```
   mrmbios -d bios.ini -a
   ```

   Daraufhin wird die folgende Aufforderung angezeigt:

   ```
   Enter the
   manufacturer of this machine:
   ```

4. Geben Sie den Namen des Herstellers ein, und drücken Sie die Eingabetaste. Daraufhin wird die folgende Aufforderung angezeigt:

   ```
   Enter the model of this machine.
   ```

5. Geben Sie das Maschinenmodell ein, und drücken Sie die Eingabetaste. Daraufhin wird die Datei bios.ini erstellt.
6. Fügen Sie die aktualisierte Datei bios.ini aller Maschinen mit Hilfe eines Texteditors zu einer Datei zusammen.
7. Sichern Sie die aktualisierte Datei bios.ini im Verzeichnis $DBDIR/inventory/INI des TMR-Servers in der TMR, von der aus Sie die Inventory-Suchläufe gestartet haben.
8. Führen Sie einen Hardwaresuchlauf auf allen PCs im Netzwerk aus, um die Tabelle PC_SYS_PARAMS im Konfigurations-Repository anhand der aktuellen Datei bios.ini zu aktualisieren.

Weitere Vorgehensweise

Sie sind nun mit den Optionen zur Anpassung von Inventory-Suchläufen und Softwaresignaturen vertraut und können jetzt mit Inventory Hardware-, Software- und Benutzerdaten erfassen. Informationen hierzu können Sie dem Kapitel „Inventory-Profile verteilen” auf Seite 197 entnehmen.

6

Mit Inventory-Profilen arbeiten

In diesem Kapitel werden die einzelnen Schritte erläutert, die beim Erstellen, Anpassen, Klonen und Löschen von Inventory-Profilen erforderlich sind. Bei der Konfiguration Ihrer Inventory-Profile sollten Sie berücksichtigen, welche Informationen erfasst werden sollen, wie diese Informationen verwendet werden und welche Einschränkungen in Ihrer Tivoli-Umgebung zu beachten sind. Nachdem Sie entschieden haben, wie Sie die von Tivoli Inventory erfassten Informationen am besten verwenden können, können Sie mit der Konfiguration der Inventory-Profile beginnen.

Weitere Informationen zur Verteilung von Inventory-Profilen und zum Ausfüllen des Konfigurations-Repository durch Tivoli Inventory können Sie dem Abschnitt „Inventory-Profile verteilen" auf Seite 197 entnehmen. Im Abschnitt „Benutzerspezifische Daten mit Tivoli Inventory erfassen" auf Seite 121 können Sie Informationen dazu finden, wie Tivoli Inventory so angepasst werden kann, dass auch nicht standardmäßig gespeicherte Daten erfasst werden.

Dieses Kapitel enthält folgende Informationen:

- Erläuterungen zur Verwendung von Inventory-Profilen. (Siehe „Inventory-Profile verwenden" auf Seite 164.)
- Informationen zum Erstellen von Inventory-Profilen. (Siehe „Inventory-Profile erstellen" auf Seite 165.)
- Informationen zum Einstellen der Suchanweisungen und weiterer Optionen in einem Inventory-Profil. (Siehe „Inventory-Profile anpassen" auf Seite 171.)

- Vorgehensweise beim Erstellen eines Inventory-Profils durch Klonen eines bereits vorhandenen Profils. (Siehe „Inventory-Profile klonen” auf Seite 192.)
- Vorgehensweise beim Löschen eines Inventory-Profils. (Siehe „Inventory-Profile löschen” auf Seite 194.)
- Vorgehensweise beim Umbenennen eines Inventory-Profils. (Siehe „Inventory-Profile umbenennen” auf Seite 196.)

Inventory-Profile verwenden

Mit Hilfe von Inventory-Profilen können Sie eine Konfigurationsdatei an einen Endpunkt verteilen, nach Hardware- und Softwaredaten suchen oder benutzerdefinierte Skripts bzw. ausführbare Dateien ausführen.

Tivoli Inventory bietet mehrere Optionen zum Anpassen von Inventory-Profilen. Sie können beispielsweise ein Inventory-Profil erstellen, dass unterschiedliche Datengruppen von PCs und UNIX-Systemen abruft. Durch diese Flexibilität ist es möglich, nur die von der jeweiligen Plattform benötigten Daten zu erfassen. Ebenso können Sie angeben, welche Hardwaredaten erfasst werden sollen, und festlegen, welche Dateien und Verzeichnisse bei den Softwaresuchläufen durchsucht werden sollen und welche nicht.

Anmerkung: An NFS angehängte Systeme können von Tivoli Inventory nicht durchsucht werden. Auf Windows NT-Systemen werden zugeordnete Laufwerke nicht von Tivoli Inventory durchsucht.

Sie können ein Profil so konfigurieren, dass auf den Zielmaschinen ein Skript ausgeführt wird. Über Skripts ist es möglich, optionale Suchprogramme auszuführen. Beispielsweise können Sie ein Skript erstellen und ausführen, das die Benutzerdaten aus einer Textdatei abruft und diese Daten dann in das Management Information Format (MIF) konvertiert. Außerdem können Sie über die Ausführung eines Skripts in Ihren Profilen die in Tivoli Inventory zur Verfügung gestellte BIOS-Wörterverzeichnisdatei aktualisieren.

Weitere Informationen zu BIOS-spezifischen Skripts finden Sie unter „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121.

Sie haben außerdem die Möglichkeit anzugeben, ob die bei einem Suchlauf erstellte MIF-Datei von dem Inventory-Profil gelesen werden soll. Mit Hilfe dieser Option können Sie ein Inventory-Profil erstellen, das den Endpunkt durchsucht und dessen Daten lokal speichert, und ein anderes Profil, das die Daten an das Konfigurations-Repository sendet.

Die Datensicherung in der RDBMS-Datenbank ist zeitraubend, wenn dabei die bereits vorhandenen Informationen durch die aktuellen Daten ersetzt werden sollen. Sie haben jedoch die Möglichkeit, nur die Daten zu sichern, die sich seit dem letzten Suchlauf geändert haben. Bei Angabe dieser Option werden von Tivoli Inventory die aktuellen MIF-Dateien mit denen verglichen, die beim letzten Suchlauf erstellt wurden; anschließend werden nur die Unterschiede im Konfigurations-Repository gespeichert.

Inventory-Profile erstellen

Inventory-Profile werden über einen Profilmanager erstellt. Mit diesem Profilmanager können Sie Ihre Profile in logische Gruppen zusammenfassen. Für die Erstellung eines Inventory-Profils können Sie entweder einen bereits vorhandenen Profilmanager verwenden oder einen neuen Profilmanager erstellen. Sollen Tivoli-Profile mit Hilfe von Profilmanagern beispielsweise nach ihrer Funktion zusammengefasst werden, können Sie für alle Inventory-Profile in einem Richtlinienbereich einen neuen Profilmanager namens ’Inventory-Profile’ erstellen. Wenn Sie dagegen über Profilmanager verfügen, die Endpunkte nach Betriebssystemen zusammenfassen, dann könnten Sie in jedem Profilmanager ein Inventory-Profil erstellen. Jedes Profil kann ein Skript bzw. eine ausführbare Datei und Suchanweisungen enthalten, die auf das Betriebssystem der Systeme im Profilmanager zugeschnitten sind.

Wenn Sie ein Inventory-Profil für die Verteilung an Endpunktsubskribenten erstellen, müssen Sie sicherstellen, dass sich der Profilmanager im datenlosen Endpunktmodus befindet. Stellen Sie bei der Erstellung eines Inventory-Profils für die Verteilung an die Subskribenten eines Profilmanagers sicher, dass die Option 'Datenloser Endpunktmodus' nicht ausgewählt ist. Vergewissern Sie sich, dass es sich bei den Subskribenten des Profilmanagers um zulässige Ziele für das Profil handelt. Als gültige Ziele für ein Inventory-Profil kommen nur Endpunkte und Profilmanager in Frage. Handelt es sich bei einigen der Subskribenten des Profilmanagers um verwaltete Knoten, werden diese von Tivoli Inventory ignoriert.

Informationen zum Zuordnen von Systemen als Subskribenten zu einem Profilmanager finden Sie im *Tivoli Management Framework Benutzerhandbuch*.

Beim Benennen von Inventory-Profilen muss verschiedenes berücksichtigt werden. Bei der Erstellung einer großen Zahl von Inventory-Profilen empfiehlt sich die Vergabe von Namen, die die Funktion des betreffenden Profils verdeutlichen. So kann beispielsweise einem Profil, das nur für die Suche nach Hardwaredaten konfiguriert ist, der Name 'Hardware_Suchlauf' zugewiesen werden.

In der folgenden Tabelle sind die für die Erstellung eines Inventory-Profils erforderlichen Berechtigungsklassen aufgeführt:

Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Inventory-Profil erstellen	Profilmanager	**senior** oder **super**

Profile können entweder über die Tivoli-Arbeitsoberfläche oder die Befehlszeile erstellt werden.

Über die Arbeitsoberfläche

Führen Sie die folgenden Schritte aus, um ein Inventory-Profil über die Tivoli-Arbeitsoberfläche zu erstellen:

1. Fügen Sie in einem Richtlinienbereich die Ressourcen **InventoryConfig** und **ProfileManager** zur Liste der verwalteten Ressourcen hinzu.

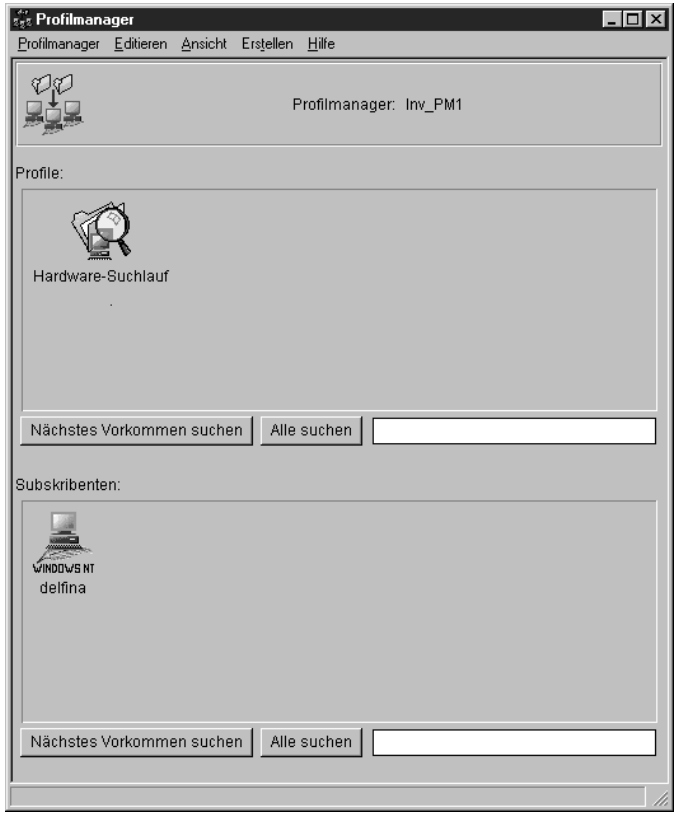
   Anweisungen zum Hinzufügen von Ressourcenarten und zum Erstellen von Profilmanagern in einem Richtlinienbereich finden Sie im *Tivoli Management Framework Benutzerhandbuch*.

2. Erstellen Sie einen Profilmanager, in dem das Inventory-Profil gespeichert wird, bzw. wählen Sie einen Profilmanager aus.

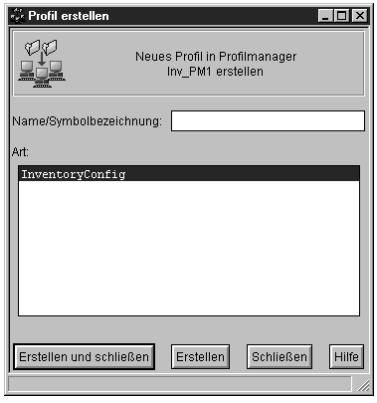
3. Ordnen Sie die Endpunkte und Profilmanager als Subskribenten zu, an die von diesem Profilmanager Profile verteilt werden sollen.

   Weitere Informationen zu Profilmanagern und Subskribenten können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

4. Klicken Sie im Richtlinienbereich doppelt auf das Symbol für einen Profilmanager. Daraufhin wird das Fenster **Profilmanager** geöffnet.



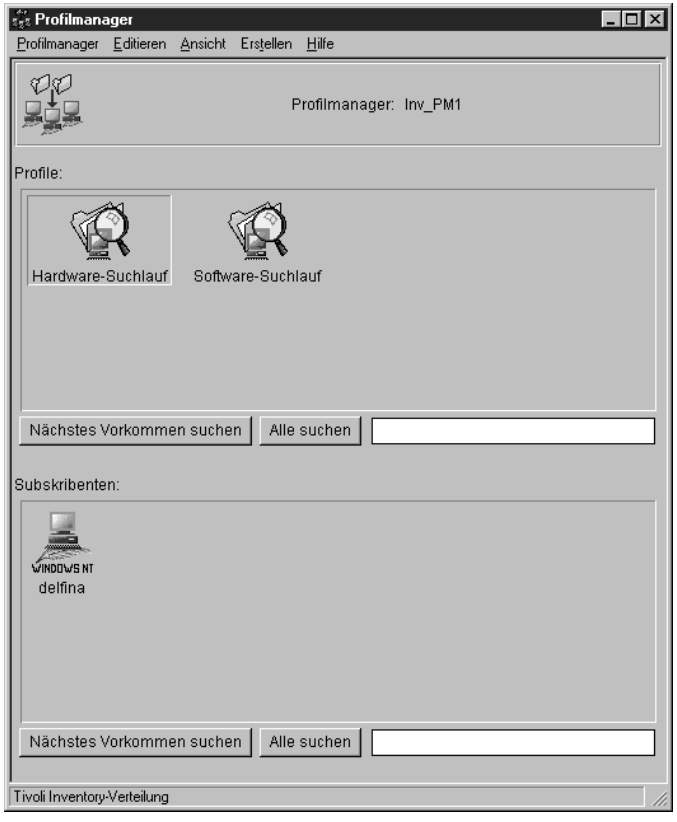
5. Wählen Sie im Fenster **Profilmanager** aus dem Menü **Erstellen** die Option **Profil** aus. Daraufhin wird der Dialog **Profil erstellen** geöffnet.



6. Geben Sie im Feld **Name/Symbolbezeichnung** einen eindeutigen Namen für das Profil ein.
7. Wählen Sie in der Liste **Art** den Eintrag **InventoryConfig** aus.

   Anmerkung: Falls nicht verfügbar, müssen Sie die Ressourcenart **InventoryConfig** Ihrem Richtlinienbereich als verwaltete Ressource hinzufügen. Weitere Informationen hierzu finden Sie im *Tivoli Management Framework Benutzerhandbuch.*

8. Klicken Sie auf die Schaltfläche **Erstellen und schließen**, um das neue Profil zu erstellen und in das Fenster **Profilmanager** zurückzukehren. Im Fenster **Profilmanager** wird nun ein Symbol für das neue Inventory-Profil angezeigt. Diesem Profil sind die Standardsuchoptionen zugeordnet. Anweisungen zum Anpassen eines Profils können Sie dem Abschnitt „Inventory-Profile anpassen” auf Seite 171 entnehmen.



Über die Befehlszeile

Inventory-Profile müssen in einem Profilmanager gespeichert werden, daher ist die Erstellung eines Richtlinienbereichs und eines Profilmanagers erforderlich, in dem das Profil erstellt wird. Hinweise zum Erstellen von Richtlinienbereichen und Profilmanagern mit den Befehlen **wcrtprf** und **wcrtprfmgr** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.

Geben Sie zum Erstellen des Profils **Enterprise** im Profilmanager **Jamaica** den Befehl **wcrtprf** wie folgt ein:

```
wcrtprf @ProfileManager:Jamaica InventoryConfig \
Enterprise
```

Dabei gilt:

@ProfileManager:Jamaica
: **Jamaica** gibt den Namen des Profilmanagers an, in dem das Profil erstellt werden soll.

InventoryConfig
: Gibt die Profilart an, die erstellt werden soll.

Enterprise
: Gibt **Enterprise** als Namen des neuen Profils an.

Weitere Informationen zum Erstellen eines Profils über die Befehlszeile finden Sie im Abschnitt zum Befehl **wcrtprf** im Handbuch *Tivoli Management Framework Reference Manual*.

Inventory-Profile anpassen

Sie haben die Möglichkeiten, Profile oder Profilgruppen an die unterschiedlichen Gegebenheiten in Ihrer Tivoli-Umgebung anzupassen. Bei der Anpassung eines Profils geben Sie an, was bei der Verteilung eines Profils auf den Zielsystemen geschehen soll. Sie können beispielsweise ein Inventory-Profil erstellen, mit dem Sie die gesamte Hardware- und Softwarekonfiguration aller Systeme in Ihrer Tivoli-Umgebung erfassen. Anschließend können Sie für nachfolgende Suchläufe ein weiteres Profil erstellen, mit dem das Konfi-

gurations-Repository nur mit neuen oder geänderten Daten zur Hard- und Software auf den betreffenden Systemen aktualisiert wird.

Beim Anpassen von Inventory-Profilen können Sie angeben, was auf dem Endpunkt passieren soll, wenn das Profil verteilt wird. Unter anderem können Sie Folgendes angeben:

- Ob ein Suchlauf erfolgen oder lediglich eine Konfigurationsdatei verteilt werden soll
- Wie die Suchlaufdaten im Konfigurations-Repository gespeichert werden sollen (alle ermittelten Daten oder nur die, die von den bereits vorhandenen abweichen)
- Die auszuführende Suchlaufart (Hardware- und/oder Softwaresuchlauf)
- Ob und welche Suchlaufergebnisse gelesen werden (Hardware-, Software- oder benutzerspezifische MIF-Dateien)
- Welche Dateien durchsucht werden
- Welche Verzeichnisse durchsucht werden
- Welche Daten erfasst werden sollen: übereinstimmende Softwaresignaturen, Header-Informationen (nur auf PC-Systemen) oder allgemeine Dateiinformationen
- Ob für die Suchläufe nach grundlegenden Dateiinformationen ein benutzerdefinierter Filter verwendet werden soll
- Ob Kontrollsummenwerte für die durchsuchten Dateien erzeugt werden sollen
- Ob Skripts, die Sie auf der Zielmaschine angeben, ausgeführt werden sollen

Allgemeine Merkmale anpassen

Die allgemeinen Merkmale eines Inventory-Profils werden über das Fenster 'Allgemeine Merkmale' auf der grafischen Benutzerschnittstelle (GUI) von Tivoli Inventory festgelegt. Zu den allgemeinen Merkmalen gehören unter anderem die Verteilungs- und Datenoptionen eines Profils. Diese Optionen gelten global für alle Suchläufe, die von einem Profil verteilt werden.

- **Verteilungsoptionen** geben an, welche Aktionen beim Verteilen eines Profils ausgeführt werden. Das Profil kann entweder mit oder ohne Ausführen eines Suchlaufs verteilt werden.

  - Bei Auswahl der Option **Konfigurationsdatei verteilen** wird die Konfigurationsdatei an den Endpunkt verteilt, der Suchlauf wird jedoch nicht gestartet. Verwenden Sie diese Option, wenn Sie die Konfigurationsdatei an einen Endpunkt verteilen wollen, der erst später durchsucht werden soll. Beispielsweise müssen Sie eine Konfigurationsdatei an einen Endpunkt verteilen, bevor Sie von einem Endpunkt aus mit dem Befehl **wepscan** einen Suchlauf einleiten.

  - Bei Auswahl der Option **Konfigurationsdatei verteilen und Suchlauf ausführen** wird die Konfigurationsdatei an den Endpunkt verteilt und der Suchlauf auf dem Endpunkt gestartet. Verwenden Sie diese Option, wenn auf dem Endpunkt sofort ein Suchlauf ausgeführt werden soll.

- **Datenoptionen** geben an, wie die Daten, die von dem Profil erfasst wurden, im Konfigurations-Repository gespeichert werden:

  - Bei Auswahl der Option **Nur anhand der Änderungen aktualisieren** werden die im Verlauf der aktuellen Verteilung gelesenen MIF-Dateien mit denen des vorangegangenen Suchlaufs verglichen und eventuelle Abweichungen gespeichert. Nur seit dem letzten Suchlauf hinzugefügte oder geänderte Daten werden im Konfigurations- Repository gespeichert. Daten aus vorherigen Suchläufen, die im aktuellen Suchlauf nicht mehr vorhanden sind, werden aus dem Konfigurations-Repository gelöscht. Die Änderungen werden nur aufgezeichnet, wenn Protokolltabellen verwendet werden. Wählen Sie diese Option im Anschluss an einen Erstsuchlauf aus, um die an das Konfigurations-Repository gesendete Datenmenge zu verringern.

  - Bei Auswahl der Option **Durch aktuelle Ergebnisse ersetzen** werden die im Konfigurations-Repository vorhandenen Daten durch die Daten ersetzt, die beim aktuellen Suchlauf erfasst wurden. Diese Option sollten Sie auswählen, wenn ein Such-

lauf zum ersten Mal durchgeführt wird und ein neues Image der auf den Zielmaschinen vorhandenen Hard- und Software gespeichert werden soll.

Wenn Sie mit Protokolltabellen arbeiten, sollten Sie beim ersten Suchlauf die Option ’Durch aktuelle Ergebnisse ersetzen’ und bei weiteren Suchläufen dann die Option ’Nur anhand der Änderungen aktualisieren’ verwenden. Weitere Informationen zu Protokolltabellen finden Sie unter „Protokollfunktion” auf Seite 366.

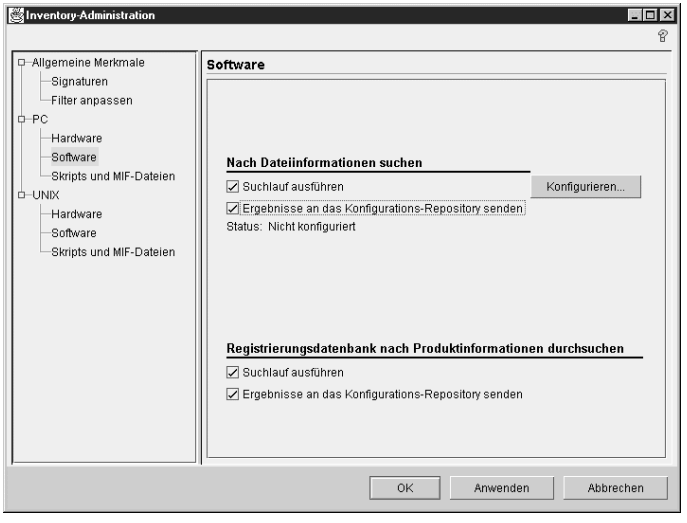
Weitere Informationen zum Fenster ’Allgemeine Merkmale’ finden Sie in der Onlinehilfe zu Tivoli Inventory.

Softwaresuchläufe anpassen

Softwaresuchläufe können über die Arbeitsoberfläche im Fenster ’Software’ angepasst werden.

Optionen für PC-Softwaresuchlauf

Wählen Sie zum Anpassen von Softwaresuchläufen auf PCs im linken Teilfenster des Fensters ’Inventory-Administration’ nacheinander **PC–>Software** aus. Daraufhin wird das Fenster ’Software’ geöffnet.



In diesem Fenster können Sie Suchläufe nach Dateiinformationen sowie Suchläufe in der Registrierungsdatenbank nach Produktinformationen aktivieren und konfigurieren.

- Mit der Option **Nach Dateiinformationen suchen** werden die auf den lokalen Festplatten eines Endpunkts gespeicherten Dateien durchsucht. Bei einem Suchlauf nach Dateiinformationen können Sie Daten wie z. B. Dateiname, Dateigröße, Pfad, Berechtigungen sowie das Datum der Erstellung, Änderung und des letzten Zugriffs erfassen. Für die durchsuchten Dateien können Sie Kontrollsummen generieren lassen. Außerdem haben Sie die Möglichkeit, bei dem Suchlauf bestimmte Dateien, Dateitypen oder Verzeichnisse herauszufiltern. Klicken Sie zum Konfigurieren dieses Suchlaufs auf **Konfigurieren**, und legen Sie dann die gewünschten Optionen in der Anzeige 'Konfiguration für Softwaresuchlauf' fest. Weitere Informationen finden Sie im Abschnitt „Optionen für die Konfiguration des Softwaresuchlaufs" auf Seite 178.

  Unter **Status** wird angezeigt, ob Sie während der aktuellen Sitzung bei der Bearbeitung des Profils die Optionen in der Anzeige 'Konfiguration für Softwaresuchlauf' geändert haben.

  Um die Option **Nach Dateiinformationen suchen** zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

  - Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
  - Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

    Sie können die beiden Optionen **Suchlauf ausführen** und **Ergebnisse an das Konfigurations-Repository senden** zusammen auswählen, um ein Profil zu erstellen, das einen Endpunkt durchsucht und direkt danach die Daten an das Konfigurations-Repository sendet.

    Alternativ dazu können Sie auch zwei Profile erstellen, wobei bei dem einen nur die Option **Suchlauf ausführen** und bei

dem anderen nur **Ergebnisse an das Konfigurations-Repository senden** ausgewählt ist. Dadurch haben Sie die Möglichkeit, den Endpunkt zu durchsuchen und die zu erfassenden Daten erst später zu speichern. Beispielsweise könnten Sie Endpunkte während der Geschäftszeiten durchsuchen und die Daten dann außerhalb der Spitzenzeiten an das Konfigurations-Repository senden, wenn eine größere Bandbreite im Netzwerk zur Verfügung steht.

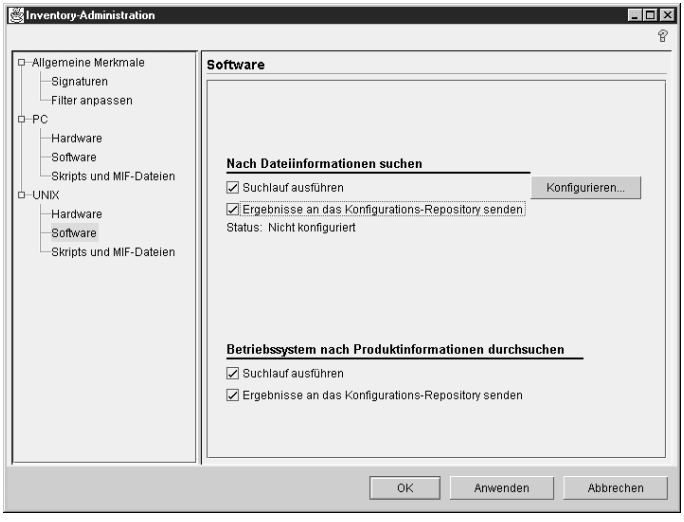
- Mit **Registrierungsdatenbank nach Produktinformationen durchsuchen** wird die Registrierungsdatenbank von Endpunkten auf unterstützten Windows-Plattformen nach Informationen zu installierten Produkten durchsucht. Um die Option **Registrierungsdatenbank nach Produktinformationen durchsuchen** zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

  - Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
  - Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

  Anmerkung: Diese Option wirkt sich nicht auf Endpunkte unter NetWare und OS/2 aus.

Optionen für UNIX-Softwaresuchlauf

Klicken Sie zum Anpassen von Softwaresuchläufen auf UNIX-Systemen auf **UNIX–>Software**. Daraufhin wird das Fenster 'Software' geöffnet.



In diesem Fenster können Sie Suchläufe nach Dateiinformationen sowie Suchläufe im Betriebssystem nach Produktinformationen aktivieren und konfigurieren.

- Mit der Option **Nach Dateiinformationen suchen** werden die auf den lokalen Festplatten eines Endpunkts gespeicherten Dateien durchsucht. Bei einem Suchlauf nach Dateiinformationen können Sie Daten wie z. B. Dateiname, Dateigröße, Pfad, Gruppe, Besitzer, Berechtigungen, Erstellungs-, Änderungs- und Zugriffsdatum sowie optional Kontrollsummenwerte erfassen. Außerdem haben Sie die Möglichkeit bei dem Suchlauf bestimmte Dateien, Dateitypen oder Verzeichnisse herauszufiltern. Unter **Status** wird angegeben, ob die Suche nach Dateiinformationen konfiguriert ist. Klicken Sie zum Konfigurieren dieses Suchlaufs auf **Konfigurieren** und legen Sie dann die gewünschten Optionen in der Anzeige 'Konfiguration für Softwaresuchlauf' fest.

Weitere Informationen finden Sie im Abschnitt „Optionen für die Konfiguration des Softwaresuchlaufs”.

Um die Option **Nach Dateiinformationen suchen** zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

- Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
- Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

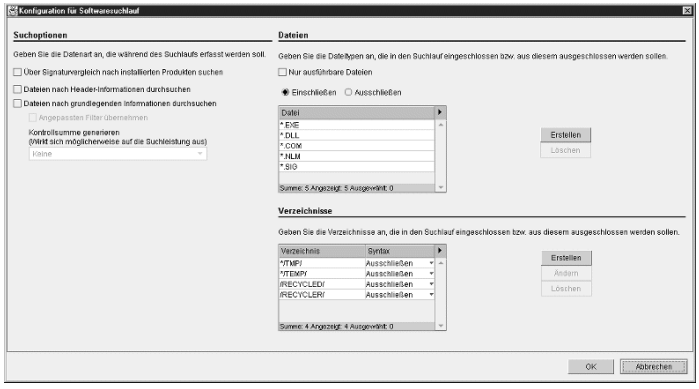
■ Mit der Option **Betriebssystem nach Produktinformationen durchsuchen** wird das UNIX-Betriebssystem nach Informationen zu den installierten Produkten und Programmkorrekturen durchsucht. Um diesen Suchlauf zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

- Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
- Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

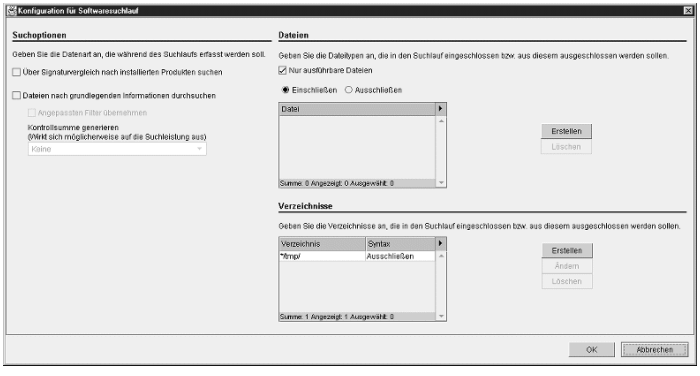
Optionen für die Konfiguration des Softwaresuchlaufs

Mit Hilfe des Dialogfensters ’Konfiguration für Softwaresuchlauf’ können Sie weitere Anpassungen an den Optionen für den Softwaresuchlauf vornehmen. Öffnen Sie das Dialogfenster ’Konfiguration für Softwaresuchlauf’, indem Sie im Fenster ’Software’ auf ’Konfigurieren’ klicken. In diesem Fenster können Sie angeben, welche Art von Informationen der Suchlauf erfassen soll und welche Dateien und Verzeichnisse in den Suchlauf eingeschlossen bzw. welche davon ausgeschlossen werden sollen.

Die folgende Abbildung zeigt das Dialogfenster ’Konfiguration für Softwaresuchlauf’ für PCs:



Die folgende Abbildung zeigt das Dialogfenster ’Konfiguration für Softwaresuchlauf’ für UNIX-Systeme:



Im folgenden Abschnitt finden Sie eine Beschreibung der Optionen im Dialogfenster ’Konfiguration für Softwaresuchlauf’.

Suchlaufoptionen

Geben Sie unter **Suchoptionen** die Datenart an, die erfasst werden soll. Jede dieser Optionen wendet eine andere Methode zum Erfassen von Daten an.

- **Über Signaturvergleich nach installierten Produkten suchen**. Bei dieser Suchlaufart wird die Liste der Softwaresignaturen an

den Endpunkt verteilt. Diese Liste wird mit den auf dem Endpunkt gesuchten Dateien verglichen. Stimmt eine Datei mit der Softwaresignatur überein, werden die Softwaresignaturdaten dieser Datei an das Konfigurations-Repository gesendet. Die Softwaresignaturdaten beinhalten Namen, Größe und normalerweise den Kontrollsummenwert der Schnellsuche der Datei, mit der das Softwareprodukt identifiziert wird. Bei dieser Identifizierungsdatei handelt es sich in der Regel um die primäre ausführbare Datei für das Produkt, z. B. notepad.exe. Benutzerdefinierte Softwaresignaturen können darüber hinaus noch weitere Daten liefern.

Suchläufe über Softwaresignaturen liefern nicht zu allen Dateien auf einem Endpunkt Daten. Die gelieferten Daten beziehen sich nur auf Dateien, die mit einer Softwaresignatur übereinstimmen. Da nur Daten für solche Dateien gesendet werden, liefert dieser Suchlauf weniger Daten als Suchläufe nach grundlegenden Dateiinformationen. Daher wird die Datenmenge reduziert, die über das Netzwerk gesendet und im Konfigurations-Repository gespeichert wird. Da bei dieser Art des Suchlaufs die einzelnen Verarbeitungsschritte zum Abgleich der Softwaresignaturen mit den durchsuchten Dateien auf den jeweiligen Endpunkten erfolgen, ist diese Methode effizienter als der Abgleich von Signaturen im Konfigurations-Repository, der mittels eines Suchlaufs nach grundlegenden Dateiinformationen durchgeführt wird. Weitere Informationen zu Softwaresignaturen finden Sie unter „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121.

- **Dateien nach Header-Informationen durchsuchen**. Ein Header ist ein Speicherbereich in einer EXE- oder DLL-Datei, in dem die Daten der Anwendung gespeichert sind, der die Datei zugeordnet ist. Mit dieser Option wird im Header der durchsuchten Dateien nach Header-Informationen gesucht. Die Header von Microsoft Windows-Dateien enthalten normalerweise folgende Daten: Firmenname, Produktname und Produktversion. Diese Option liefert nicht nur Informationen zu Dateien mit Übereinstimmungen bei den Softwaresignaturen, sondern zu allen durchsuchten Dateien, die Header-Informationen enthalten.

Bei Softwaresuchläufen auf UNIX-Systemen ist diese Option nicht verfügbar. Diese Option wirkt sich nicht bei Suchläufen auf Endpunkten unter NetWare und OS/2 aus.

- **Dateien nach grundlegenden Informationen durchsuchen**. Bei dieser Option werden Name und Pfad von durchsuchten Dateien erfasst. Auf manchen Plattformen werden noch weitere Informationen erfasst, wie z. B. Erstellungsdatum, Änderungsdatum und Datum des letzten Zugriffs. Im Gegensatz zu Suchläufen nach Softwaresignaturen oder Header-Informationen kann diese Art von Suchlauf Informationen zu allen Dateien auf den durchsuchten Endpunkten erfassen. Ebenso können Sie einen Filter auf den Suchlauf anwenden, so dass nur Informationen zu angegebenen Dateien erfasst werden.

  Durch Aktivierung des Kontrollkästchens 'Angepassten Filter übernehmen' können Sie einen Filter auf Suchläufe nach grundlegenden Informationen anwenden. Wenn dieses Kontrollkästchen aktiviert ist, werden solche Suchläufe auf die Dateien beschränkt, die im Dialogfenster 'Filter anpassen' aufgelistet sind. Weitere Informationen zum Fenster 'Filter anpassen' finden Sie in der Onlinehilfe zu Tivoli Inventory.

  Die Option 'Kontrollsumme generieren' gibt Ihnen die Möglichkeit, den Kontrollsummenwert für die durchsuchten Dateien zu erfassen. Bei der algorithmischen Verarbeitung von Kontrollsummen wird der Inhalt einer Datei mit einem einzelnen festen Wert verknüpft. Durch Vergleichen dieser Werte können Sie ermitteln, ob sich der Inhalt einer Datei im Laufe der Zeit, z. B. aufgrund beschädigter Daten oder eines Virus, geändert hat. Beispielsweise können Sie die MD5-Kontrollsummenwerte aller Dateien in Ihrem Unternehmen zusammenstellen und dann überwachen, ob sich der Wert ändert, um festzustellen, ob sich evtl. ein "Trojanisches Pferd" eingeschlichen hat. Folgende Kontrollsummenoptionen stehen zur Verfügung:

  - Die Option **Keine** inaktiviert die Kontrollsummenfunktion für das Inventory-Profil, das Sie gerade anpassen.

- Die Option **Schnelle CRC-Prüfung** verwendet den CRC-32-Algorithmus, um einen 32-Bit-Wert auf der Basis des ersten Kilobyte jeder Datei zu erzeugen. Dies ist die schnellste Option für Kontrollsummen, sie erzeugt aber den am wenigsten zuverlässigen Wert.
- Die Option **Vollständige CRC-Prüfung** verwendet den CRC-32-Algorithmus, um einen 32-Bit-Wert auf Basis des vollständigen Inhalts der Dateien zu erzeugen. Diese Option ist langwieriger als die Option 'Schnelle CRC-Prüfung', sie ergibt aber einen zuverlässigeren Wert.
- Die Option **MD5** erzeugt einen 128-Bit-Wert auf der Basis des vollständigen Inhalts der Dateien. Der von der Option 'MD5' gelieferte Wert ist zuverlässiger als der Wert der schnellen oder der vollständigen CRC-Prüfung, seine Erzeugung benötigt jedoch mehr Zeit.

Dateien

Geben Sie mit Hilfe der Liste 'Dateien' die Dateinamen oder Dateitypen an, die in den Suchlauf eingeschlossen bzw. aus ihm ausgeschlossen werden sollen. Sie können die Suchzeit verkürzen, indem Sie die Dateien einschränken, die vom Profil durchsucht werden. Sie können die Namen von Dateien angeben, die einzuschließen oder auszuschließen sind, oder aber durch die Angabe der Dateierweiterungen (z. B. *.exe) ein- bzw. auszuschließende Dateitypen festlegen. Diese Liste wird auf Suchläufe nach übereinstimmenden Softwaresignaturen, Suchläufen nach Header-Informationen und Suchläufen nach grundlegenden Dateiinformationen angewendet.

Aktivieren Sie die Option 'Nur ausführbare Dateien', wenn ausschließlich nach ausführbaren Dateien gesucht werden soll. Auf Windows-Systemen sind ausführbare Dateien als Dateien mit einer der folgenden Erweiterungen definiert: EXE, COM, CMD oder BAT. Bei UNIX-Systemen sind solche Dateien als ausführbare Dateien definiert, bei denen die Ausführungsberechtigung gesetzt ist.

Verzeichnisse

Geben Sie in der Liste 'Verzeichnisse' die Verzeichnisse an, die in den Suchlauf eingeschlossen bzw. aus diesem ausgeschlossen werden sollen. Sie können gleichzeitig sowohl einzuschließende als auch auszuschließende Verzeichnisse angeben. Sie können die Suchzeit verkürzen, indem Sie die Verzeichnisse einschränken, die vom Profil durchsucht werden. Sie können entweder bestimmte auszuschließende Verzeichnisse auflisten oder aber einen Teilpfad angeben, so dass alle Unterverzeichnisse unterhalb dieses Pfades ausgeschlossen werden. Diese Liste wird auf Suchläufe nach übereinstimmenden Softwaresignaturen, Suchläufen nach Header-Informationen und Suchläufen nach grundlegenden Dateiinformationen angewendet.

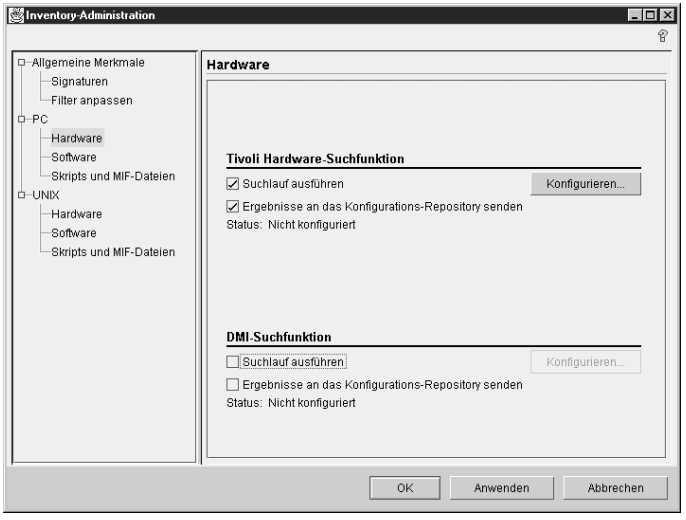
Weitere Informationen zum Fenster 'Konfiguration für Softwaresuchlauf' finden Sie in der Onlinehilfe zu Tivoli Inventory.

Hardwaresuchläufe anpassen

Tivoli Inventory sucht PC- und UNIX-Systeme standardmäßig nach bestimmten Hardwarekomponenten ab. Sie können jedoch ein Inventory-Profil so anpassen, dass es nur die von Ihnen ausgewählten Hardwarekomponenten sucht. Des Weiteren können Sie das Inventory-Profil für Hardwaresuchläufe auf PCs so konfigurieren, dass die Hardwaredaten über die DMI (Desktop Management Interface)-Ebene eines PCs erfasst werden, und Sie können die DMI-Attribute auswählen, die in die Suche eingeschlossen werden sollen. Hardwaresuchläufe können über die Arbeitsoberfläche im Fenster 'Hardware' angepasst werden.

PC-Hardwaresuchläufe

Wählen Sie zum Anpassen von Hardwaresuchläufen auf PCs im linken Teilfenster des Fensters 'Inventory-Administration' nacheinander **PC–>Hardware** aus. Daraufhin wird das Fenster 'Hardware' geöffnet.



In diesem Fenster können Sie die Tivoli Hardware-Suchfunktion, die DMI-Suchfunktion oder beide aktivieren und konfigurieren.

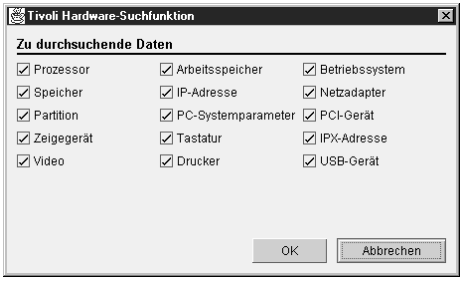
Tivoli Hardware-Suchfunktion

Die Tivoli Hardware-Suchfunktion verwendet eine Kombination aus Aufrufen an Hardware, System und Registrierungsdatenbank, um die Kenndaten zu Hardwarekomponenten abzurufen. Bei einigen Windows-Betriebssystemen beinhaltet die Suchfunktion die Unterstützung für CIM (Common Information Model) und WMI (Windows Management Instrumentation), um normierte Daten abzurufen.

Um diesen Suchlauf zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

- Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
- Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

Bei Tivoli Inventory können Sie angeben, nach welchen Hardwareattributen gesucht werden soll. Beispielsweise können Sie ein Inventory-Profil erstellen, das nur nach der installierten Speicherkapazität sucht. Klicken Sie zum Anpassen der Liste der Daten, die von der Tivoli Hardware-Suchfunktion erfasst werden, auf **Konfigurieren**. Daraufhin wird das Dialogfenster 'Tivoli Hardware-Suchfunktion' geöffnet.



Weitere Informationen zum Anpassen von PC-Hardwaresuchläufen finden Sie in der Onlinehilfe zu Tivoli Inventory.

DMI-Suchfunktion

Mit der DMI-Suchfunktion wird die DMI-Ebene eines PCs nach Hardwareinformationen durchsucht. Bei DMI handelt es sich um einen Industriestandard für die Beschreibung und den Zugriff auf Informationen zu den verschiedenen PCs und PC-Komponenten. Manche PCs enthalten eine DMI-Serviceebene, in der Informationen zu dem PC enthalten sind, auf dem sie installiert ist. Mit Hilfe des DMI-Standards können Sie die PC-Hardware unabhängig von Maschinentyp, Betriebssystem oder Verwaltungssystem verwalten.

Beispielsweise können Sie von allen Systemen, auf denen vom Hersteller die DMI-Ebene installiert wurde, die Seriennummer abfragen. Je nach Hersteller enthält die DMI-Ebene unter Umständen auch Informationen zu den mit Ihrer Hardware verbundenen Kosten (Erwerb, Wartung, Strom).

Bei Ausführung des DMI-Suchprogramms werden von Tivoli Inventory die Attribute aus der auf den Zielsystemen installierten DMI-Ebene abgerufen und in einer MIF-Datei (dmiscan.mif) gespeichert.

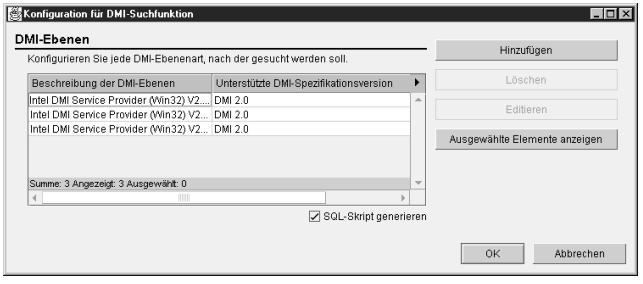
Tivoli Inventory liest diese MIF-Datei und schreibt entsprechende Daten in das Konfigurations-Repository.

Folgende Konfigurationsschritte sind zur Suche nach DMI-Informationen erforderlich:

1. Aktivieren Sie den DMI-Suchlauf im Fenster 'Hardware', indem Sie **Suchlauf ausführen** und ggf. **Ergebnisse an das Konfigurations-Repository senden** auswählen:

   - Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
   - Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

2. Klicken Sie auf **Konfigurieren**, um den Dialog 'Konfiguration für DMI-Suchfunktion' aufzurufen.



Konfigurieren Sie in diesem Dialogfenster die DMI-Optionen für einen PC-Hardwaresuchlauf. Bevor Sie eine DMI-Ebene durchsuchen können, müssen Sie eine Liste mit den Hardwareattributen erstellen, nach denen gesucht werden soll. Rufen Sie dazu jede zu durchsuchende DMI-Ebenenart ab, und bearbeiten Sie diese. Weitere Informationen zur Verwendung dieses Dialogfensters finden Sie in der Onlinehilfe zu Tivoli Inventory.

Anmerkungen:

a. Stellen Sie vor dem Versuch, eine DMI-Ebene von einem Endpunkt abzurufen, sicher, dass Sie ein Inventory-Profil mindestens ein Mal an diesen Endpunkt verteilt haben. Ein Inventory-Profil verteilt unabhängig von seiner Konfiguration folgende Dateien an den Endpunkt: dmiscan.exe, wcdmi.dll und wdmiutil.dll. Diese Dateien müssen auf dem Endpunkt vorhanden sein, bevor Sie die DMI-Ebene abrufen können.

b. Rufen Sie dazu jede zu durchsuchende DMI-Ebenenart ab, und konfigurieren Sie diese. Die DMI-Ebenen unterscheiden sich je nach Hersteller und Modell.

3. Aktivieren Sie das Kontrollkästchen **SQL-Skript generieren** im Dialogfenster 'Konfiguration für DMI-Suchfunktion'. Durch Auswahl dieser Option generiert Tivoli Inventory automatisch ein SQL-Skript, mit dem Sie anschließend im Konfigurations-Repository angepasste Tabellen für die DMI-Suchlaufdaten erstellen können.

4. Klicken Sie im Dialogfenster 'Konfiguration für DMI-Suchfunktion' auf **OK**, um das Skript zu generieren. Dieses Skript trägt den Namen dmi_*RDBMS_type*_schema.sql, wobei *RDBMS_type* für den RDBMS-Typ steht, in dem das Konfigurations-Repository gespeichert ist. Das Skript befindet sich im Verzeichnis $BINDIR/TME/INVENTORY auf dem TMR-Server. Führen Sie dieses Skript aus, bevor Sie ein Profil verteilen, mit dem DMI-Suchlaufdaten an das Konfigurations-Repository gesendet werden.

5. Stellen Sie, um das Skript im Konfigurations-Repository auszuführen, eine Client-Verbindung zum Datenbankserver her (z. B. eine SQL*Plus-Sitzung), auf dem das Konfigurations-Repository von Tivoli Inventory gespeichert ist, und führen Sie dann das Skript aus. Weitere Informationen zum Ausführen eines Skripts über eine Client-Verbindung finden Sie unter „Konfigurations-Repository erstellen" auf Seite 63. Dieses Kapitel enthält Anweisungen zur Installation des Schemas für das Konfigurations-Repository. Für jeden unterstützten RDBMS-Typ ist ein eigener Abschnitt enthalten. Verwenden Sie den Ihrem Konfigurations-

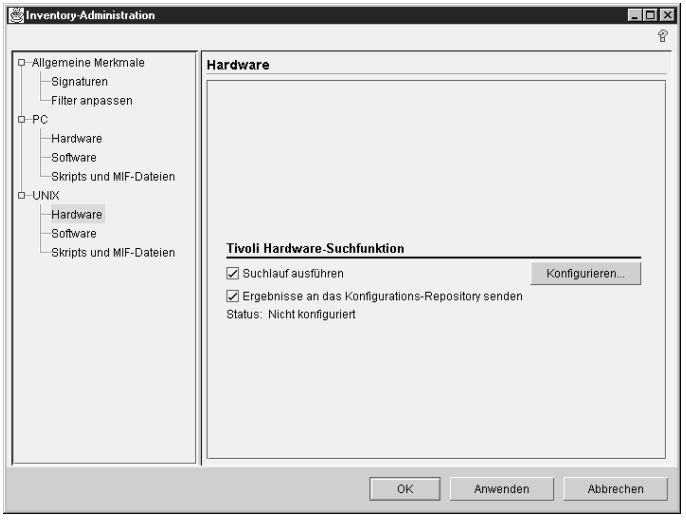
Repository entsprechenden Abschnitt. Befolgen Sie die Anweisungen in diesem Abschnitt, verwenden Sie dabei aber anstelle von inv_*xxx*_schema.sql das Skript dmi_*RDBMS_type*_schema.sql. Achten Sie bei der Ausführung dieses Skripts auf Fehlernachrichten, die darauf hinweisen, dass das Skript nicht ordnungsgemäß beendet wurde.

6. Verteilen Sie das Profil an die Zielsysteme.

Weitere Informationen zum Konfigurieren von DMI-Suchlaufoptionen finden Sie in der Onlinehilfe von Tivoli Inventory. Hilfe bei der Fehlersuche bei DMI-Suchläufen finden Sie unter „Fehlerbehebung” auf Seite 511.

UNIX-Hardwaresuchläufe

Klicken Sie zum Anpassen von Hardwaresuchläufen auf UNIX-Systemen auf **UNIX–>Hardware**. Daraufhin wird das Fenster ’Hardware’ geöffnet.



In diesem Fenster können Sie die Tivoli Hardware-Suchfunktion aktivieren und konfigurieren. Um diesen Suchlauf zu aktivieren, müssen Sie eine oder beide der folgenden Optionen auswählen:

- Mit **Suchlauf ausführen** wird der Endpunkt durchsucht, und die erfassten Daten werden in einer MIF-Datei auf dem Endpunkt gespeichert.
- Mit der Option **Ergebnisse an das Konfigurations-Repository senden** wird die MIF-Datei vom Endpunkt abgerufen, und die Daten werden an das Konfigurations-Repository gesendet.

Klicken Sie zum Anpassen der Liste der Daten, die von der Tivoli Hardware-Suchfunktion erfasst werden, auf **Konfigurieren**. Daraufhin wird das Dialogfenster 'Tivoli Hardware-Suchfunktion' geöffnet.

Weitere Informationen zum Anpassen von UNIX-Hardwaresuchläufen finden Sie in der Onlinehilfe zu Tivoli Inventory.

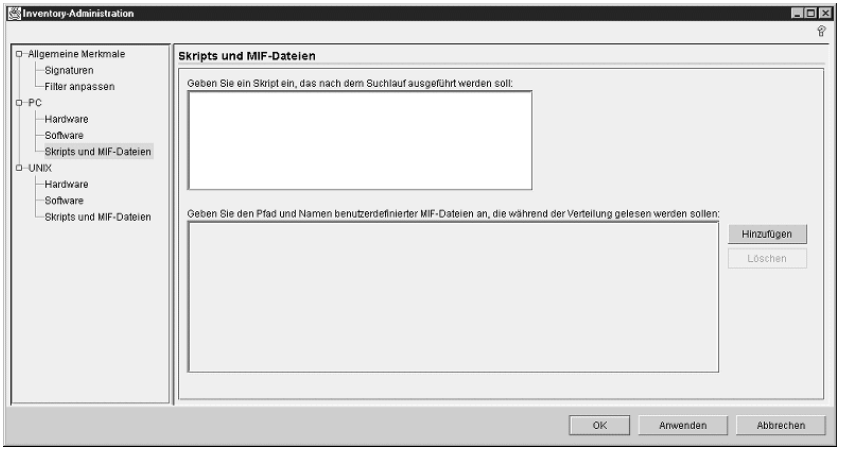
Benutzerdefinierte Skripts und MIF-Dateien verwenden

Sie können ein Inventory-Profil so anpassen, dass während einer Profilverteilung benutzerdefinierte Skripts ausgeführt werden. Die Skripts werden nach dem Suchlauf, jedoch vor dem Lesen der MIF-Dateien ausgeführt. Mit dieser Option können Sie MIF-Benutzerdateien erstellen oder eine MIF-Datei ändern, die bei einem Suchlauf erstellt wurde. Anweisungen zur Verwendung von benutzerdefinierten MIF-Dateien können Sie unter „Benutzerspezifische Daten mit Tivoli Inventory erfassen" auf Seite 121 finden.

Die verwendeten Skripts müssen unter dem Betriebssystem der jeweiligen Zielsysteme ausführbar sein. Sie können z. B. ein Shell-Skript für UNIX-Ziele, eine BAT-Datei für Windows-Ziele oder eine CMD-Datei für OS/2-Ziele schreiben. Diese Skripts müssen die Verzeichnisse für alle Befehle oder Dateien enthalten, auf die in ihnen verwiesen wird.

Anmerkung: Auf NetWare-Endpunkten können keine Skripts ausgeführt werden.

Wenn Sie benutzerdefinierte Skripts oder MIF-Dateien für PC-Systeme hinzufügen wollen, wählen Sie **PC->Skripts und MIF-Dateien** aus. Wählen Sie zum Hinzufügen von benutzerdefinierten Skripts oder MIF-Dateien für UNIX-Systeme **UNIX->Skripts und MIF-Dateien** aus. Daraufhin wird das Fenster 'Skripts und MIF-Dateien' geöffnet.



Geben bzw. fügen Sie im oberen Textfeld ein Stapel- oder Befehlsskript ein, das nach einem Inventory-Suchlauf ausgeführt werden soll. Geben Sie in diesem Feld das vollständige Skript ein, nicht den Pfad zu einer Skriptdatei. Listen Sie im unteren Textfeld alle benutzerdefinierten MIF-Dateien auf, die bei einer Verteilung gelesen werden sollen. Klicken Sie auf **Hinzufügen**, um eine Datei zur Liste hinzuzufügen. Wenn Sie eine Datei aus der Liste löschen wollen, klicken Sie erst in diesem Textfeld auf die betreffende Datei und dann auf **Löschen**.

Weitere Informationen zur Verwendung von Skripts und benutzerdefinierten MIF-Dateien finden Sie in der Onlinehilfe zu Tivoli Inventory.

Berechtigungsklassen

Profile können entweder über die Tivoli-Arbeitsoberfläche oder die Befehlszeile angepasst werden. In der folgenden Tabelle sind die für das Anpassen eines Inventory-Profils erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Profil anpassen	Tivoli Inventory-Profil	**Inventory_edit**, **admin**, **senior** oder **super**

Anmerkung: Wenn ein Benutzer der Berechtigungsklasse **Inventory_edit** ein Inventory-Profil über die Tivoli-Arbeitsoberfläche anpassen will, muss er außerdem in dem Richtlinienbereich, in dem das Profil gespeichert ist, über die Berechtigungsklasse **user** verfügen.

Über die Arbeitsoberfläche

Das Verfahren zum Anpassen eines Inventory-Profils über die Tivoli-Arbeitsoberfläche können Sie der Onlinehilfe zu Tivoli Inventory entnehmen.

Über die Befehlszeile

Zur Anpassung eines Inventory-Profils über die Befehlszeile können Sie zunächst mit einem oder mehreren der folgenden Befehle Informationen zu den aktuellen Einstellungen des Profils abrufen: **wgetinvglobal**, **wgetinvpcfiles**, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixhw** und **wgetinvunixsw**. Anschließend können Sie zum Ändern dieser Einstellungen einen oder mehrere der folgenden Befehle verwenden: **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvpcsw**, **wsetinvunixfiles**, **wsetinvunixhw** und **wsetinvunixsw**. Unter „Befehle” auf Seite 239 finden Sie weitere Informationen zu diesen Befehlen und zu ihrer Verwendung beim Anpassen von Inventory-Profilen.

Inventory-Profile klonen

Sie haben die Möglichkeit, eine Kopie eines Inventory-Profils zu erstellen, indem Sie es klonen. (Dazu muss das Profil jedoch umbenannt werden.) Dies ist besonders hilfreich bei PCs in Ihrer Tivoli-Umgebung, deren Dateien sich auf verschiedenen Laufwerken befinden. Sie können das Originalprofil klonen und anschließend den Laufwerkbuchstaben für die Dateien ändern, nach denen gesucht werden soll.

In der folgenden Tabelle sind der Kontext und die zum Klonen erforderlichen Berechtigungsklassen aufgeführt:

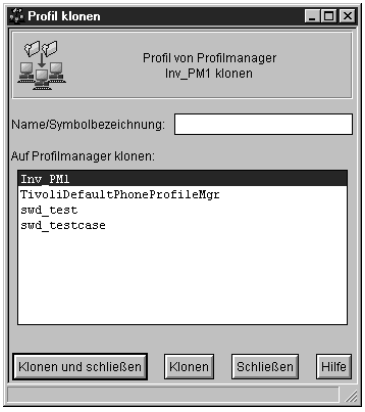
Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Profil klonen	Profilmanager	**senior** oder **super**

Sie können diese Task entweder über die Tivoli-Arbeitsoberfläche oder über die Befehlszeile ausführen.

Über die Arbeitsoberfläche

Führen Sie die folgenden Schritte aus, um ein Inventory-Profil über die Tivoli-Arbeitsoberfläche zu klonen:

1. Wählen Sie aus einem Profilmanager das Profil aus, das geklont werden soll.
2. Wählen Sie **Profile->Klonen** im Menü **Editieren** aus. Daraufhin wird der Dialog **Profil klonen** geöffnet.



3. Geben Sie im Feld **Name/Symbolbezeichnung** den Namen des neuen Profils ein.
4. Wählen Sie in der Liste **Auf Profilmanager klonen** den Profilmanager aus, in dem das neue Profil gespeichert werden soll.
5. Klicken Sie auf die Schaltfläche **Klonen und schließen**, um eine Kopie des Profils zu erstellen und anschließend in das Fenster **Profilmanager** zurückzukehren.

Über die Befehlszeile

Um das Inventory-Profil **UNIX_First_Scan** zu klonen, müssen Sie den Befehl **wcrtprf** wie folgt eingeben:

```
wcrtprf -c @UNIX_First_Scan \
@ProfileManager:First_Scan_Profiles InventoryConfig UNIX_Update
```

Dabei gilt:

-c @UNIX_First_Scan
: Gibt **UNIX_First_Scan** als das Profil an, das geklont werden soll. Für die Angabe des Profilnamens müssen Sie das 'at'-Zeichen (@) verwenden.

@ProfileManager:First_Scan_Profiles
: Gibt an, dass das neue Inventory-Profil im Profilmanager **First_Scan_Profiles** gespeichert wird.

InventoryConfig
Gibt die Profilart an, die geklont werden soll.

UNIX_Update
Gibt an, dass dem neuen Profil der Name **UNIX_Update** zugeordnet wird. Die Angabe eines Objektpfads (**/Regions/...**) oder registrierten Namens (**@name**) ist nicht zulässig. Sie müssen einen neuen Namen angeben.

Weitere Informationen zum Klonen von Inventory-Profilen finden Sie in der Beschreibung des Befehls **wcrtprf** im Handbuch *Tivoli Management Framework Reference Manual*.

Inventory-Profile löschen

Beim Löschen wird ein Profil aus der Tivoli-Objektdatenbank und das entsprechende Symbol aus dem Profilmanager entfernt.

Anmerkung: Wenn Sie ein Inventory-Profil löschen, für das Sie eine Verteilung geplant haben, müssen Sie den Job aus dem Planer löschen. Der Job wird nicht automatisch beim Entfernen des entsprechenden Profils entfernt. Weitere Informationen zur Verwendung des Planers finden Sie im *Tivoli Management Framework Benutzerhandbuch*.

In der folgenden Tabelle sind der Kontext und die zum Löschen eines Profils erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Inventory-Profil löschen	Profilmanager	**senior** oder **super**

Inventory-Profile können entweder über die Tivoli-Arbeitsoberfläche oder die Befehlszeile gelöscht werden.

Über die Arbeitsoberfläche

Führen Sie die folgenden Schritte aus, um ein Inventory-Profil über die Tivoli-Arbeitsoberfläche zu löschen:

1. Wählen Sie in einem Profilmanager die Symbole der Profile aus, die Sie löschen wollen, und wählen Sie anschließend **Profile->Löschen** im Menü **Editieren** aus. Daraufhin wird der Dialog **Profile löschen** geöffnet.

   Anmerkung: Nachdem ein Profil gelöscht wurde, kann es nicht wiederhergestellt werden.

   Ignorieren Sie die Nachricht “Die Originalprofile UND alle ihre Kopien werden gelöscht”, da sie für Inventory-Profile nicht zutrifft.

2. Klicken Sie auf **Löschen**, um den Löschvorgang zu bestätigen. Das Profil wird daraufhin von Tivoli Inventory gelöscht.

Über die Befehlszeile

Sie können das Profil **UNIX_First_Scan** löschen und das entsprechende Symbol aus dem Profilmanager **First_Scan_Profiles** im Richtlinienbereich **Inventory** entfernen, indem Sie den Befehl **wdel** wie folgt eingeben:

```
wdel \
/Regions/Inventory/First_Scan_Profiles/UNIX_First_Scan
```

oder

```
wdel @InventoryConfig:UNIX_First_Scan
```

Dabei gilt:

/Regions/Inventory/First_Scan_Profiles/UNIX_First_Scan
: Gibt den Objektpfad des Inventory-Profils an, das gelöscht werden soll. Sie können den Objektpfad bzw. den registrierten Namen von Tivoli-Objekten wie z. B. Profilen oder Profilmanagern angeben.

Weitere Informationen zum Befehl **wdel** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*. Weitere Informationen zu Objektpfaden finden Sie unter „Befehle” auf Seite 239.

Inventory-Profile umbenennen

Gehen Sie zum Umbenennen eines Inventory-Profils folgendermaßen vor:

1. Klonen Sie das Profil. Geben Sie dem Profil den gewünschten Namen des Originalprofils.
2. Löschen Sie das Originalprofil.

Weitere Vorgehensweise

Die Erstellung und Anpassung von Inventory-Profilen ist damit abgeschlossen, und Sie können jetzt Suchläufe für die Systeme in Ihrer Tivoli-Umgebung durchführen. Anweisungen zur Verteilung von Inventory-Profilen an Subskribenten sowie eine Beschreibung der von Tivoli Inventory erfassten Daten können Sie dem Kapitel „Inventory-Profile verteilen” auf Seite 197 entnehmen. Weitere Informationen zur Anpassung von Tivoli Inventory finden Sie unter „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121.

7

Inventory-Profile verteilen

Nach der Erstellung und Anpassung der Inventory-Profile können Sie diese an Endpunkte in Ihrer Tivoli-Umgebung verteilen. In diesem Kapitel werden die einzelnen Schritte bei der Verteilung von Inventory-Profilen über die Tivoli-Arbeitsoberfläche und über die Befehlszeilenschnittstelle erläutert. Falls Sie noch keine Inventory-Profile erstellt haben, holen Sie dies jetzt nach; Anweisungen hierzu finden Sie im Kapitel „Mit Inventory-Profilen arbeiten" auf Seite 163.

Dieses Kapitel enthält folgende Informationen zu Inventory-Suchläufen:

- Eine Ablaufbeschreibung von Inventory-Suchläufen sowie eine Beschreibung der Vorgehensweise bei einer Verteilung von Inventory-Profilen. (Siehe hierzu „Inventory-Profile verteilen" auf Seite 198.)
- Informationen dazu, wie ein Suchlauf von einem Endpunkt aus eingeleitet werden kann, ohne dass er von einem Inventory-Profil aus verteilt werden muss. (Siehe hierzu „Von Endpunkten initiierten Suchlauf ausführen" auf Seite 204.)
- Eine Beschreibung, wie die erfassten Daten im Konfigurations-Repository gespeichert werden. (Siehe hierzu „Daten im Konfigurations-Repository speichern" auf Seite 206.)

Inventory-Profile verteilen

Sie können das Inventory-Profil über die Tivoli-Arbeitsoberfläche oder die Befehlszeilenschnittstelle an die Zielmaschinen verteilen.

Der erste Suchlauf auf den Maschinen in Ihrer Umgebung kann unter Umständen einige Zeit in Anspruch nehmen. Nach dem ersten Suchlauf können Sie aber für die weiteren Läufe ein Inventory-Profil verwenden, das die Ergebnisse des aktuellen Suchlaufs mit denen des vorhergehenden Suchlaufs vergleicht.

Bei Auswahl der Option 'Nur anhand der Änderungen aktualisieren' im Fenster 'Allgemeine Merkmale' speichert Tivoli Inventory nur neue oder geänderte Daten im Konfigurations-Repository. Da in diesem Fall normalerweise weniger Daten anfallen, wird der Sicherungsvorgang beschleunigt, d. h. nachfolgende Suchläufe nehmen wesentlich weniger Zeit in Anspruch als der erste Suchlauf. Informationen zur Einstellung von Suchläufen können Sie dem Kapitel „Mit Inventory-Profilen arbeiten" auf Seite 163 entnehmen.

Tivoli Inventory liefert Daten zu Ereignissen und Fehlern, die eventuell bei einer Inventory-Profilverteilung aufgetreten sind. In der Standardeinstellung werden diese Daten an die Hinweisgruppe von Tivoli Inventory gesendet. Durch Verwendung des Befehls **wsetinvglobal** mit der Option **–l** können Sie Tivoli Inventory so konfigurieren, dass diese Daten an eine Protokolldatei, die TEC-Konsole, an die Tivoli Inventory-Hinweisgruppe oder eine Kombination dieser Zieladressen gesendet werden. Weitere Informationen zum Befehl **wsetinvglobal** können Sie dem Abschnitt „Befehle" auf Seite 239 entnehmen. Weitere Informationen zur Anzeige von Gruppenhinweisen finden Sie im *Tivoli Management Framework Benutzerhandbuch*. Solche Hinweise werden von Tivoli Inventory unter anderem in den folgenden Fällen gesendet:

- Ein Suchlauf verläuft erfolglos, weil ein Endpunkt nicht erreichbar ist.
- Fehler bei einer RDBMS-Operation.

- Eine MIF-Benutzerdatei konnte in dem angegebenen Pfad nicht gefunden werden.
- Eine MIF-Benutzerdatei liegt nicht im richtigen Format vor.

In der folgenden Tabelle sind die für die Verteilung eines Inventory-Profils erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Berechtigungsklasse
Inventory-Profile verteilen	Inventory-Profil	**Inventory_scan**, **senior** oder **super**

Anmerkung: Wenn ein Benutzer der Berechtigungsklasse **Inventory_scan** ein Inventory-Profil über die Tivoli-Arbeitsoberfläche verteilen will, muss er außerdem in dem Richtlinienbereich, in dem das Profil gespeichert ist, über die Berechtigungsklasse **user** verfügen.

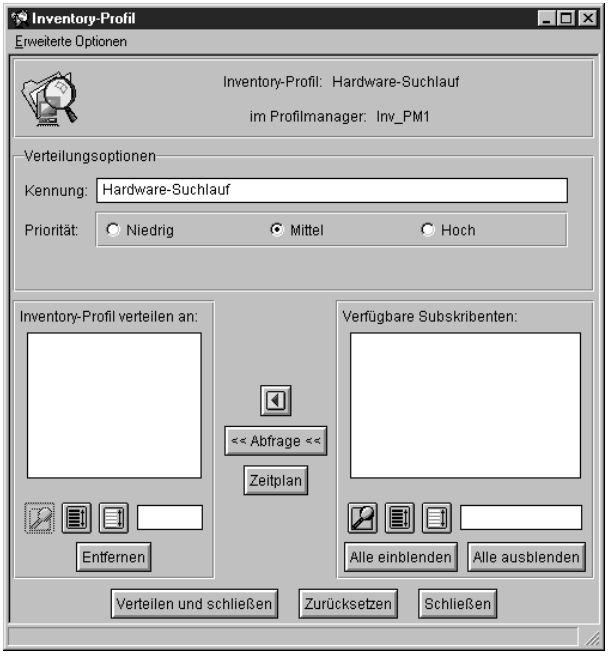
Profile können über die Tivoli-Arbeitsoberfläche oder die Befehlszeile verteilt werden.

Mit MDist 2 können Sie Profilverteilungen Prioritäten zuweisen. Sie können zwischen hoher, mittlerer und niedriger Priorität auswählen. Die Verarbeitung der Verteilungsprioritäten wird gesteuert, indem einer Verteilung eine Anzahl von Verbindungen zugewiesen wird, die ein Repeater öffnen kann, um eine Verteilung einer bestimmten Priorität durchzuführen. Darüber hinaus versucht der Repeater, vorübergehend eine für Verteilungen mit niedriger Priorität zugewiesene Verbindung zu verwenden, wenn für die gegebene Priorität keine Verbindungen zur Verfügung stehen. Weitere Informationen darüber, wie einer Verteilung eine Priorität zugewiesen wird und wie die maximale Anzahl von Verbindungen für eine Priorität festgelegt wird, finden Sie im *Tivoli Management Framework Benutzerhandbuch*.

Über die Arbeitsoberfläche

Führen Sie die folgenden Schritte aus, um ein Inventory-Profil über die Tivoli-Arbeitsoberfläche zu verteilen.

1. Klicken Sie in einem Profilmanager mit der rechten Maustaste auf das Symbol für ein Inventory-Profil, und wählen Sie die Option **Verteilen** aus. Daraufhin wird das Dialogfenster 'Inventory-Profil' geöffnet.



2. Legen Sie die Priorität der Verteilung fest, indem Sie eins der Optionsfelder **Niedrig**, **Mittel** oder **Hoch** auswählen.

3. Wählen Sie in der Liste 'Verfügbare Subskribenten' die Maschinen oder Profilmanager aus, an die das Profil verteilt werden soll. Die Liste 'Verfügbare Subskribenten' enthält die Subskribenten des Profilmanagers, in dem das Profil gespeichert ist.

4. Sie haben folgende Möglichkeiten:
   - Verschieben Sie mit Hilfe der Pfeiltaste die gewünschten Ziele aus der Liste ’Verfügbare Subskribenten’ in die Liste ’Inventory-Profil verteilen an’.
   - Klicken Sie auf die Schaltfläche **Abfrage**, um die Ziele über eine Abfrage auszuwählen. Weitere Informationen zur Verwendung von Abfragen für die Auswahl der Zielmaschinen können Sie dem Kapitel „Inventory-Daten abfragen” auf Seite 209 entnehmen.

5. Optional können Sie auch noch die Zeitlimiteinstellungen für dieses Profil ändern. Klicken Sie hierzu auf das Menü **Erweiterte Optionen** und wählen Sie darin die Option **Zeitlimiteinstellungen** aus. Das Dialogfenster ’Zeitlimiteinstellungen’ wird daraufhin geöffnet.

In diesem Dialog können Sie folgende Optionen einstellen:

Termin

Der Termin, zu dem eine Verteilung ausläuft. Ist bis zu diesem Termin keine Verteilung an ein Ziel erfolgt, macht MDist 2 keine weiteren Verteilungsversuche. Der Standardwert für diese Option ist 72 Stunden nach der aktuellen Zeit.

Benachrichtigungsintervall

Das Intervall, in dem Massendaten, die bei einer Verteilung zurückgegeben werden, von einem Repeater an den nächsten gesendet werden. Der Standardwert ist 1 Minute.

Zeitlimit für Senden

Das Zeitlimit für Daten, die vom Server an eine Zielmaschine gesendet werden (z. B. ein Repeater, der ein Inventory-Profil an einen Endpunkt verteilt). Jedes Mal, wenn ein Paket über das Netzwerk gesendet wird, wird das Zeitlimit für das betreffende Paket auf diesen Wert festgelegt. In der Standardeinstellung ist kein Zeitlimit gesetzt.

Zeitlimit für Ausführung

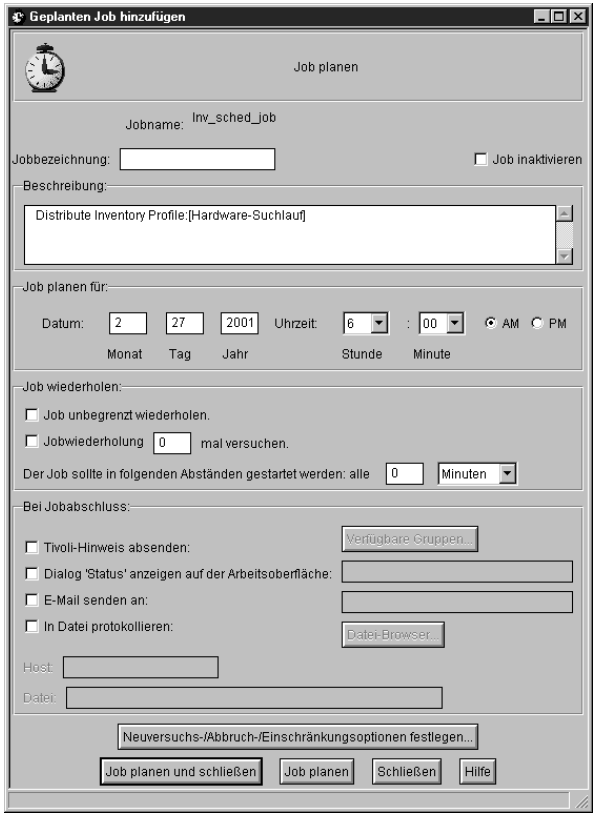
Die Zeit, die ein Kollektor darauf wartet, dass ein Endpunkt die Ergebnisse zurückgibt (die Rückgaben auf abwärts gerichtete Aufrufe), nachdem alle Daten gesendet wurden. Dieser Wert entspricht dem Schlüsselwort für den Leistungsparameter *execute_timeout*, der mit Hilfe des Befehls **wmdist** und der Option **–s** definiert wurde. (Weitere Informationen zum Befehl **wmdist** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.) Wenn Sie vorhaben, einen langwierigen Suchlauf auszuführen, weil beispielsweise sehr viele Laufwerke durchsucht werden müssen, müssten Sie für diesen Wert eine entsprechend lange Zeit festlegen. Standardmäßig ist kein Zeitlimit gesetzt.

6. Sie haben folgende Möglichkeiten:

   - Klicken Sie auf **Verteilen und schließen**, um das Profil an die ausgewählten Ziele zu verteilen. Wenn die Verteilung an eines der Ziele fehlschlägt, wird ein Dialogfenster angezeigt, in dem die betreffenden Subskribenten aufgeführt sind.

- Klicken Sie auf **Zeitplan**, um festzulegen, dass die Verteilung an ein Ziel im Falle eines Fehlers automatisch wiederholt werden soll.

  Daraufhin wird der Dialog **Geplanten Job hinzufügen** geöffnet. Weitere Informationen zu Tivoli Scheduler finden Sie im *Tivoli Framework Benutzerhandbuch*.



Über die Befehlszeile

Um das Inventory-Profil zu verteilen, müssen Sie den Befehl **wdistinv** wie folgt eingeben:

```
wdistinv @InventoryConfig:First_Scan \
@Endpoint:sly @Endpoint:robbie
```

Dabei gilt:

@InventoryConfig:First_Scan
Gibt den Namen des Inventory-Profils an.

@Endpoint:sly @Endpoint:robbie
Gibt an, dass das Profil an die Endpunkte **sly** und **robbie** verteilt werden soll.

Von Endpunkten initiierten Suchlauf ausführen

Mit Tivoli Inventory kann ein Suchlauf von einem Endpunkt aus initiiert werden, ohne dass er von einem Inventory-Profil aus verteilt werden muss. Diese Funktion bietet zahlreiche Verwendungsmöglichkeiten. Sie können beispielsweise von Endpunkten initiierte Suchläufe verwenden, um folgende Funktionen auszuführen:

- Eine Maschine bei jedem Neustart durchsuchen.
- Ein Laptop-System jedes Mal durchsuchen, wenn sich ein mobiler Benutzer am Netzwerk anmeldet.
- Eine Maschine durchsuchen, die vom Netzwerk getrennt ist.

Einen von einem Endpunkt initiierten Suchlauf können Sie automatisch mit Hilfe eines Skripts oder manuell mit dem Befehl **wepscan** ausführen. Weitere Informationen zum Befehl **wepscan** können Sie dem Abschnitt „wepscan” auf Seite 271 entnehmen.

Wenn Sie einen Suchlauf auf einem Endpunkt einleiten wollen, müssen Sie zuerst eine Konfigurationsdatei an den Endpunkt verteilen. Sie können ein Profil erstellen, mit dessen Hilfe eine Konfigurationsdatei an einen Endpunkt verteilt wird, ohne dass der Endpunkt durchsucht wird. Wählen Sie hierzu im Fenster ’Allgemeine Merkmale’ die Option ’Konfigurationsdatei verteilen’ aus. Weitere Informationen zum Erstellen eines Inventory-Profils finden Sie unter „Mit Inventory-Profilen arbeiten” auf Seite 163.

Das folgende Verfahren dient als Demonstrationsbeispiel für eine Verwendungsmöglichkeit des vom Endpunkt initiierten Suchlaufs. Durch dieses Verfahren wird auf einem Windows NT-Endpunkt jedes Mal, wenn ein Benutzer sich anmeldet, ein Suchlauf gestartet. Für die Art des vom Endpunkt initiierten Suchlaufs müssen zwei Tasks ausgeführt werden: Konfigurieren der Endpunktumgebung und Einleiten des Suchlaufs. Im vorliegenden Beispiel werden diese beiden Tasks zu einer Stapeldatei zusammengefasst, die dann jedes Mal ausgeführt wird, wenn der Benutzer sich anmeldet.

Um diese Art eines vom Endpunkt initiierten Suchlaufs einzurichten, müssen Sie folgende allgemeine Arbeitsschritte ausführen:

1. Erstellen Sie eine Stapeldatei ähnlich dem folgenden Beispiel:

```
@ECHO OFF
if %SYSTEMROOT%/==/ set SYSTEMROOT=%WINDIR%
if %SYSTEMROOT%/==/ goto bad
CALL %SYSTEMROOT%\TIVOLI\LCF\1\lcf_env.cmd
if %LCFROOT%/==/ goto bad
if not exist %LCFROOT%\inv\SCAN\wepscan.exe goto notfound
%LCFROOT%\inv\SCAN\wepscan
goto quit
:bad
echo Could not source environment. Scan aborted.
goto quit
:notfound
echo Could not find Tivoli Inventory Endpoint Scanner.
:quit
echo Done.
```

2. Stellen Sie diese Stapeldatei auf Ihrem Anmeldeserver in ein gemeinsam benutztes Verzeichnis oder auf den lokalen Computer, der bei der Anmeldung durchsucht werden soll.

3. Starten Sie den Benutzer-Manager von Windows NT. Öffnen Sie im Dialogfenster ’Benutzereigenschaften’ die Anzeige ’Umgebungsprofil für Benutzer’. Geben Sie im Textfeld ’Anmeldeskriptname’ den vollständigen Pfad und Namen der Stapeldatei ein.

Daten im Konfigurations-Repository speichern

Die bei einem Suchlauf erfassten Daten werden von Tivoli Inventory im Konfigurations-Repository gespeichert. Das Konfigurations-Repository enthält Tabellen, in denen die Daten gespeichert werden, die bei der Verteilung von Inventory-Profilen erfasst wurden. Der Inhalt der Tabellen im Konfigurations-Repository sowie deren Beziehung untereinander wird durch das Konfigurations-Repository-Schema festgelegt. Eine grafische Darstellung dieses Schemas finden Sie auf dem Poster mit dem Tivoli Inventory-Konfigurations-Repository, das mit dem vorliegenden Handbuch geliefert wird. Eine Aufstellung aller Tabellen, Anzeigen und Abfragen, die in Tivoli Inventory zur Verfügung stehen, können Sie dem Abschnitt „Konfigurations-Repository” auf Seite 365 entnehmen.

Bei Hardware- bzw. Softwaresuchläufen auf Zielmaschinen werden von Tivoli Inventory Daten zu einer vorgegebenen Gruppe von Hardware- und Softwarekomponenten aus den MIF-Dateien, die von der Suchsoftware erstellt wurden, erfasst. Sie können bei der Verteilung von Inventory-Profilen auch benutzerspezifische Daten erfassen und speichern, die bei einem Hardware- oder Softwaresuchlauf nicht erfasst werden. Informationen zur Verwendung von MIF-Benutzerdateien sowie zur Erweiterung des Konfigurations-Repositorys können Sie dem Kapitel „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121 entnehmen. Anweisungen zur Erstellung neuer Tabellen sowie entsprechende Beispielprozeduren finden Sie im Abschnitt „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen” auf Seite 141. Anweisungen zur Erstellung neuer Ansichten sowie entsprechende Beispielskripts können Sie dem Abschnitt „Benutzerdaten anzeigen” auf Seite 156 entnehmen.

Weitere Vorgehensweise

Der Suchlauf auf den Systemen in Ihrer Tivoli-Umgebung ist damit abgeschlossen, und Sie können jetzt die dabei erfassten Daten aus dem Konfigurations-Repository von Tivoli Inventory überprüfen. Informationen zur Erstellung und Verwendung von Abfragen können Sie dem Kapitel „Inventory-Daten abfragen” auf Seite 209 entnehmen.

8

Inventory-Daten abfragen

In diesem Kapitel werden die Vorgänge erläutert, die bei der Suche nach Informationen im Konfigurations-Repository erforderlich sind. Die Abfragefunktion von Tivoli Management Framework ermöglicht Ihnen die Erstellung von SQL-Anweisungen, mit deren Hilfe Sie die Daten abrufen können, die über die Verteilung von Inventory-Profilen erfasst wurden.

Die Abfragefunktion umfasst Abfragebibliotheken und Abfragen. Die Abfragebibliotheken befinden sich in Richtlinienbereichen und enthalten die Abfragen. Mit jeder Abfrage wird eine bestimmte Datenkategorie aus einer bestimmten Ansicht oder Tabelle im Konfigurations-Repository abgerufen. In Tivoli Inventory stehen mehrere vordefinierte Abfragen zur Verfügung. Darüber hinaus haben Sie die Möglichkeit, eigene Abfragen zu erstellen. Im vorliegenden Kapitel werden die verschiedenen Einsatzmöglichkeiten von Abfragen für den Abruf von Inventory-Daten aus dem Konfigurations-Repository vorgestellt, unter anderem folgende:

- Verwendung der in Tivoli Inventory vordefinierten Abfragen. (Siehe „Inventory- und Subskriptionsabfragen verwenden” auf Seite 210.)
- Beispiel für die Erstellung eigener Abfragen über den Dialog ’Abfrage erstellen’. (Siehe „Abfragen erstellen” auf Seite 211.)
- Ausführung eigener Abfragen sowie Anzeige bzw. Sicherung der Ergebnisse. (Siehe „Abfragen ausführen” auf Seite 221.)

- Vorgehensweise bei der Verwendung von Abfragen zur Auswahl von Zielmaschinen für die Verteilung von Inventory-Profilen. (Siehe „Subskribenten über Abfragen ermitteln” auf Seite 225.)
- Vorgehensweise bei der Verwendung von Abfragen für den Abruf von Inventory-Daten zu einer bestimmten Maschine in einer TMR. (Siehe „Informationen zu einzelnen Clients abrufen” auf Seite 229.)
- Abfragen für miteinander verbundene TMRs erstellen und Daten jeder einzelnen TMR anzeigen. (Siehe „Abfragen im TMR-Verbund verwenden” auf Seite 233.)

Eine umfassende Beschreibung der Abfragefunktionen und -befehle von Tivoli Management Framework können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

Inventory- und Subskriptionsabfragen verwenden

In Tivoli Inventory stehen Ihnen zwei Gruppen von Abfragen zur Verfügung, mit denen ein Zugriff auf die mit Tivoli Inventory ermittelten Daten im Konfigurations-Repository möglich ist. Mit den Inventory-Abfragen können verschiedene Gruppen von Hardware- und Softwareinformationen aus dem Konfigurations-Repository abgerufen werden. Beispielsweise erhalten Sie bei Ausführung der Abfrage INVENTORY_HWARE grundlegende Hardwareinformationen zu allen vorhandenen Maschinen.

Von den in Tivoli Inventory vorhandenen Subskriptionsabfragen dagegen werden Listen mit Maschinen zurückgegeben, die bestimmten Suchkriterien entsprechen. Beispielsweise erhalten Sie bei Ausführung der Abfrage AIX_SUBSCRIPTION eine Liste mit den Maschinen, auf denen das Betriebssystem AIX installiert ist.

Die von Tivoli Inventory zur Verfügung gestellten Abfragebibliotheken und Abfragen werden bei der Installation des Produkts erstellt. Weitere Informationen über die Ausführung der Skripts inventory_query.sh, subscription_query.sh, h_inventory_query.sh und h_subscription_query.sh zur Erstellung von Inventory- und

Subskriptionsabfragen finden Sie unter „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87.

Abfragen erstellen

Sie haben die Möglichkeit, zum Abrufen der gewünschten Daten aus dem Konfigurations-Repository eigene Abfragen zu erstellen und anzupassen.

Bei der Erstellung einer Abfrage müssen Sie das Repository, das durchsucht werden soll, sowie die gewünschten Informationen angeben. Daher sollten Sie vor Erstellung einer Abfrage zunächst festlegen, welche Daten abgerufen werden sollen. Anschließend müssen Sie die Tabellen und Spalten im Konfigurations-Repository ermitteln, die diese Informationen enthalten, und diese in der Abfrage angeben. Im Poster mit dem Tivoli Inventory-Konfigurations-Repository ist das Datenmodell des Konfigurations-Repository dargestellt. Eine kurze Beschreibung zu den einzelnen Tabellen finden Sie unter „Konfigurations-Repository” auf Seite 365.

Wenn die Tabellen und Spalten, die bei der Abfrage durchsucht werden sollen, bekannt sind, können Sie die Ansicht auswählen, die diese Spalten enthält. Eine Liste der Ansichten können Sie dem „Konfigurations-Repository” auf Seite 365 entnehmen. Falls keine Ansicht alle von Ihnen gewünschten Spalten enthält, können Sie eine neue Ansicht erstellen. Anweisungen hierzu können Sie dem Abschnitt „Benutzerdaten anzeigen” auf Seite 156 entnehmen.

Bei der Erstellung einer Abfrage müssen Sie im Dialog ’Abfrage erstellen’ folgende Felder ausfüllen:

- **Abfragename**: Dabei muss es sich um einen eindeutigen Namen in der TMR handeln.
- **Repository**: Diese Angabe legt fest, welche Tabellen und Ansichten bei der Abfrage durchsucht werden.
- **Tabellen-/Anzeigenname**: Über die Angabe der Tabelle bzw. Ansicht im Repository werden die Spalten festgelegt, die bei der Abfrage durchsucht werden sollen.

- **Ausgewählte Spalten**: Gibt die Spalten innerhalb der Tabelle bzw. Ansicht an, die durchsucht werden sollen.

Sie können außerdem für die Abfrage eine Beschreibung sowie eine oder mehrere WHERE-Klauseln angeben. Dazu geben Sie im Bereich WHERE-Klausel des Dialogs eine SQL-Suchklausel an, die die Daten festlegt, die von der Abfrage zurückgegeben werden. Die Beispielabfrage im vorliegenden Abschnitt veranschaulicht die Verwendung mehrerer Optionen der Abfragefunktion.

Eine umfassende Beschreibung der Abfragefunktion von Tivoli Management Framework können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

In der folgenden Tabelle sind die dafür erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Abfrage erstellen	Abfragebibliothek	**Query_edit**, **admin**, **senior** oder **super**

Sie können diese Abfrage entweder über die Tivoli-Arbeitsoberfläche oder über die Befehlszeile erstellen.

Über die Arbeitsoberfläche

Das folgende Beispiel zeigt die Erstellung einer Abfrage über die Tivoli-Arbeitsoberfläche. Mit der dabei erstellten Abfrage werden die folgenden Informationen aus dem Konfigurations-Repository abgerufen:

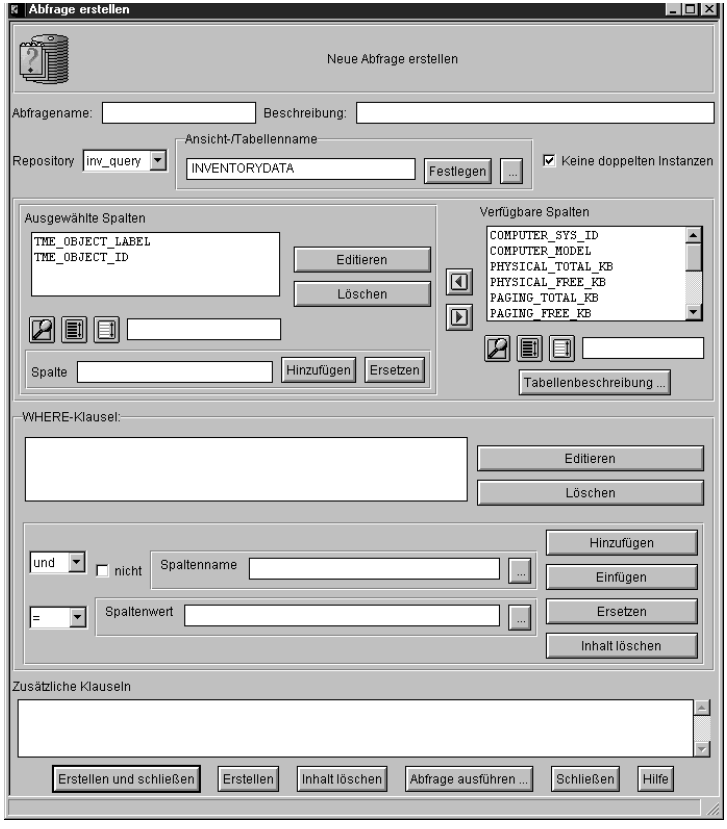
- Alle Maschinen, auf denen Microsoft Word installiert ist und die über einen Arbeitsspeicher (RAM) von mindestens 32 MB verfügen.
- Die Version von Microsoft Word, die auf den einzelnen Maschinen installiert ist.

Folgende Schritte sind zur Erstellung einer Abfrage über die Tivoli-Arbeitsoberfläche erforderlich:

1. Doppelklicken Sie im Richtlinienbereich auf das Symbol der Abfragebibliothek, die die Abfrage enthalten soll. Daraufhin wird das Fenster 'Abfragebibliothek' geöffnet.

   Anmerkung: Abfragen müssen in Abfragebibliotheken erstellt werden. Dazu können Sie entweder eine eigene Abfragebibliothek erstellen oder eine bereits vorhandene auswählen, in der die Abfrage dann gespeichert wird.

2. Wählen Sie aus dem Menü **Erstellen** die Option **Abfrage** aus. Daraufhin wird der Dialog **Abfrage erstellen** geöffnet.



3. Geben Sie im Feld **Abfragename** einen eindeutigen Namen für die Abfrage ein (z. B. MSWord_Finder). Der Name kann aus einer beliebigen Folge von alphanumerischen Zeichen in Groß- oder Kleinschreibung bestehen. Leerzeichen sind zulässig.
4. Geben Sie im Feld **Beschreibung** eine kurze Beschreibung der Abfrage ein. Angaben in diesem Feld sind optional. Als Beschreibung der vorliegenden Abfrage können Sie 'Ruft die Version von MSWord ab' angeben.
5. Wählen Sie aus der Liste **Repository** eine RDBMS-Datenbank aus, die bei der Abfrage durchsucht werden soll. Hier sind die Namen der RIM-Objekte in der lokalen TMR aufgeführt. Da bei dieser Abfrage die Ergebnisse von Suchläufen durchsucht werden, müssen Sie den Eintrag **inv_query** auswählen, um Zugriff auf das Konfigurations-Repository von Inventory zu erhalten.
6. In dem Feld für den Ansicht-/Tabellennamen haben Sie folgende Möglichkeiten:
   - Geben Sie den Namen der Ansicht INVENTORYDATA ein, und klicken Sie auf **Festlegen**.

     Anmerkung: Sie können in diesem Feld mehrere Ansichten bzw. Tabellen auswählen. Die Namen der einzelnen Tabellen bzw. Ansichten müssen durch ein Komma (,) getrennt werden. Den Spaltennamen in der Liste **Verfügbare Spalten** sind jeweils die Namen der von Ihnen ausgewählten Tabelle bzw. Ansicht vorangestellt.

- Klicken Sie auf die Schaltfläche mit den Auslassungszeichen (**...**); daraufhin wird eine Liste der im Repository vorhandenen Ansichten zur Auswahl angeboten. In dieser Liste werden die Ansichten angezeigt, die in der Tabelle QUERY_VIEWS im Konfigurations-Repository aufgeführt sind. Für diese Abfrage werden die Ansichten INVENTORYDATA und INST_SWARE_VIEW angegeben, da diese die Spalten mit den Daten zur Software sowie zum physischen Speicher enthalten.

Anmerkung: Bei Änderung des Eintrags im Feld für den Tabellen- und Ansichtnamen werden in der Liste **Verfügbare Spalten** die entsprechenden Spalten angezeigt; der Inhalt der restlichen Felder in diesem Dialog wird gelöscht, unter anderem die Einträge in der Liste **Ausgewählte Spalten** sowie in den Feldern **WHERE-Klausel** und **Zusätzliche Klauseln**.

7. Wählen Sie die Option **Keine doppelten Instanzen** aus, wenn bei der Abfrage gefundene identische Einträge im Ergebnis nur einmal angeführt werden sollen.
8. Wählen Sie in der Liste **Verfügbare Spalten** die Spalten aus, die durchsucht werden sollen, und verschieben Sie diese über den Linkspfeil in die Liste **Ausgewählte Spalten**.

Anmerkung: Unter Umständen sind einige Spaltennamen in der Liste nicht in ihrer ganzen Länge sichtbar. In diesem Fall können Sie das Listenfenster **Verfügbare Spalten** erweitern, indem Sie auf den linken bzw. rechten Rand des Fensters klicken und dieses auf die gewünschte Breite ziehen.

Im vorliegenden Beispiel sollen Informationen zur Software und zum Arbeitsspeicher abgefragt werden, daher müssen Sie die folgenden Spalten auswählen:

- INST_SWARE_VIEW.SWARE_DESC
- INST_SWARE_VIEW.SWARE_VERS
- INVENTORYDATA.PHYSICAL_TOTAL_KB

Da von dieser Abfrage eine Liste mit Maschinen zurückgegeben wird, sollten Sie außerdem noch die Spalte TME_OBJECT_LABEL angeben (dies erleichtert die Identifizierung der zurückgegebenen Maschinen):

9. Erstellen Sie im Bereich **WHERE-Klausel** eine SQL-Suchklausel; diese gibt an, welche Daten von der Abfrage zurückgegeben werden sollen. Sie können die Suchklausel über die Felder **Spaltenname** und **Spaltenwert** und die Operatorschaltflächen erstellen.

   Folgende Schritte sind zur Erstellung der ersten SQL-Anweisung erforderlich, die dem Feld **WHERE-Klausel** hinzugefügt werden soll:

   a. Die Angabe der Software, nach der gesucht werden soll, kann auf zwei Wegen erfolgen: Geben Sie SWARE_DESC im Feld 'Spaltenname' ein.

      -ODER-

      Klicken Sie auf die Schaltfläche mit den Auslassungszeichen (**...**), um die Liste 'Verfügbare Spalten' anzuzeigen, und wählen Sie hier den Eintrag SWARE_DESC aus.

   b. Wählen Sie den logischen Operator aus, um eine Verknüpfung zwischen dem Eintrag im Feld 'Spaltenname' und dem Eintrag im Feld 'Spaltenwert' herzustellen. In diesem Beispiel wird als logischer Operator das Gleichheitszeichen (=) übernommen, da über das Feld **Spaltenwert** die Software angegeben wird, nach der gesucht werden soll. (Wenn nach Maschinen ohne eine bestimmte Software gesucht werden soll, aktivieren Sie das Kontrollkästchen 'nicht'.)

c. Klicken Sie im Bereich 'Spaltenwert' auf die Schaltfläche mit den Auslassungszeichen (**...**). Wählen Sie den Eintrag **Microsoft Word** aus, und klicken Sie auf die Schaltfläche **Schließen**, da mit dieser Abfrage nach Maschinen gesucht werden soll, deren Maschinen-ID die Software Microsoft Word zugeordnet ist. Damit ist die Angabe der Suchkriterien für diese WHERE-Klausel abgeschlossen.

Anmerkung: Falls die ausgewählte Tabelle bzw. Ansicht eine große Anzahl von Zeilen enthält, kann es unter Umständen etwas dauern, bevor der Dialog 'Spaltenwert' angezeigt wird.

d. Klicken Sie auf die Schaltfläche **Hinzufügen**, um die Suchkriterien in das Feld 'WHERE-Klausel' zu übernehmen.

10. Soll das Suchergebnis weiter eingegrenzt und sollen nur die Maschinen zurückgegeben werden, die über einen Arbeitsspeicher von mindestens 32 MB verfügen, müssen Sie dem Feld 'WHERE-Klausel' die nächste SQL-Anweisung hinzufügen. Gehen Sie hierzu wie folgt vor:

    a. Geben Sie im Feld 'Spaltenname' die Spalte 'PHYSICAL_TOTAL_KB' ein; diese Spalte enthält Informationen zum physischen Speicher.

    b. Wählen Sie aus dem Menü links neben dem Feld **Spaltenwert** den logischen Operator >= aus; von der Abfrage werden dadurch alle Maschinen zurückgegeben, deren Arbeitsspeicher größer-gleich dem angegebenen Wert ist.

    c. Geben Sie im Feld **Spaltenwert** den Arbeitsspeicher ein, über den die Maschinen mindestens verfügen sollen; das ist im vorliegenden Beispiel der Wert **32000 KB**. (Speicherangaben müssen in KB erfolgen.)

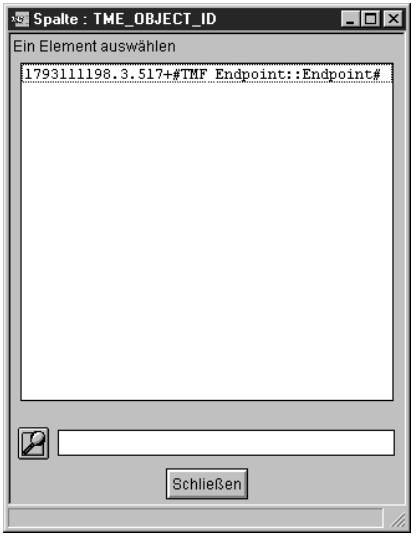
    d. Klicken Sie auf die Schaltfläche **Hinzufügen**, um die zweite WHERE-Klausel in das Feld **WHERE-Klausel** zu übernehmen.

11. Die letzte WHERE-Klausel für diese Abfrage gibt an, dass die Abfrage nur Informationen zu Maschinen in einer bestimmten TMR zurückgeben soll. Um die Abfrage auf Maschinen innerhalb einer bestimmten TMR zu beschränken, müssen Sie angeben, dass nur Daten der Maschinen zurückgegeben werden sollen, deren TME_OBJECT_ID die Nummer der betreffenden TMR enthält. Gehen Sie hierzu wie folgt vor:

    a. Geben Sie TME_OBJECT_ID im Feld **Spaltenname** ein.

    b. Wählen Sie aus dem Menü neben dem Feld **Spaltenwert** den logischen Operator **LIKE** aus. Dies ermöglicht Ihnen die Verwendung eines Platzhalterzeichens (%) bei der Angabe der Tivoli-Objekt-ID, nach der in der Abfrage gesucht werden soll.

    c. Klicken Sie auf die Schaltfläche mit den Auslassungszeichen (**...**). Daraufhin wird der Dialog **Spalte: TME_OBJECT_ID** geöffnet, in dem die TME_OBJECT_ID der einzelnen Maschinen im Konfigurations-Repository aufgeführt ist. Bei der ersten Nummer in der TME_OBJECT_ID einer Maschine handelt es sich um die Bereichsnummer der TMR, in der sich das betreffende Objekt befindet.



d. Wählen Sie im Dialog **Spalte: TME_OBJECT_ID** die TME_OBJECT_ID einer Maschine aus, die die Bereichsnummer der von Ihnen angegebenen TMR enthält, und klicken Sie auf **Schließen**. Die Tivoli-Objekt-ID wird daraufhin in das Feld **Spaltenwert** übernommen.

   Informationen zur Bestimmung der numerischen ID der TMR finden Sie unter „Abfragen im TMR-Verbund verwenden" auf Seite 233. Anweisungen zur Anzeige von Informationen zu einem bestimmten Client können Sie dem Abschnitt „Informationen zu einzelnen Clients abrufen" auf Seite 229 entnehmen.

e. Ersetzen Sie die TME_OBJECT_ID im Feld **Spaltenwert** durch die gewünschte TME-Nummer mit einem Platzhalterzeichen (z. B. **105620229%**). Hierzu müssen Sie in der TME_OBJECT_ID den Teil nach der TMR-Nummer löschen und stattdessen ein Prozentzeichen (%) angeben. Von der Abfrage werden daraufhin nur Informationen zu Maschinen in der TMR zurückgegeben, die dieser Nummer entspricht. Dadurch werden bei der Abfrage alle Clients in der angegebenen TMR berücksichtigt.

f. Klicken Sie auf **Hinzufügen**. Die letzte Klausel wird im Feld **WHERE-Klausel** angezeigt.

12. Fügen Sie die folgende Klausel dem Feld 'Zusätzliche Klauseln' hinzu:

```
INVENTORYDATA.COMPUTER_SYS_ID =
INST_SWARE_VIEW.COMPUTER_SYS_ID
```

Anmerkung: Sollen mehrere Tabellen bzw. Ansichten abgefragt werden, müssen Sie diese Tabellen bzw. Ansichten über den Bereich **Zusätzliche Klauseln** miteinander verknüpfen.

13. Klicken Sie auf **Erstellen und schließen**; daraufhin wird die Abfrage erstellt, und Sie kehren in das Abfragebibliotheksfenster zurück;

    -ODER-

    Klicken Sie auf die Schaltfläche **Erstellen**, um die Abfrage zu erstellen.

14. Klicken Sie auf die Schaltfläche **Abfrage ausführen,** um die Abfrage zu starten. Weitere Informationen zum Ausführen von Abfragen und Speichern der Suchergebnisse können Sie dem Abschnitt „Abfragen ausführen” auf Seite 221 entnehmen.

Über die Befehlszeile

In dem folgenden Beispiel wird dieselbe Abfrage (**MSWord_Finder**) wie im vorhergehenden Abschnitt erstellt, diesmal jedoch mit dem Befehl **wcrtquery**:

```
wcrtquery -d "Sucht nach der Version von MSWord" \
-r inv_query \
-v INVENTORYDATA \
-v INST_SWARE_VIEW \
-c INST_SWARE_VIEW.SWARE_DESC -c INST_SWARE_VIEW.SWARE_VERS \
-c INST_SWARE_VIEW.TME_OJBECT_LABEL -c INVENTORYDATA.PHYSICAL_TOTAL_KB \
-w "(INST_SWARE_VIEW.SWARE_DESC='Microsoft Word') und \
(INVENTORYDATA.PHYSICAL_TOTAL_KB>=32000) und \
(TME_OBJECT_ID like '1947203709%')" INVENTORY_QUERY MSWord_Finder
```

Weitere Informationen zum Befehl **wcrtquery** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Abfragen ausführen

Nach Erstellung einer Abfrage zum Abrufen von Informationen zu Ihrer Tivoli-Umgebung können Sie diese Abfrage ausführen und das Ergebnis anzeigen bzw. in einer Datei speichern. Sie haben folgende Möglichkeiten zum Ausführen einer Abfrage und zum Anzeigen bzw. Speichern der Suchergebnisse:

- Klicken Sie mit der rechten Maustaste auf das Symbol der Abfrage, und wählen Sie **Abfrage ausführen** aus.
- Klicken Sie im Dialog **Abfrage erstellen** auf **Abfrage ausführen**.
- Klicken Sie im Dialog **Abfrage editieren** auf **Abfrage ausführen**.
- Geben Sie den Befehl **wrunquery** ein.

In der folgenden Tabelle sind die dafür erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Abfrage ausführen	Der Richtlinienbereich, der die Abfragebibliothek der gültigen Abfrage enthält	**senior**, **super**, **Query_execute** oder **Query_edit**

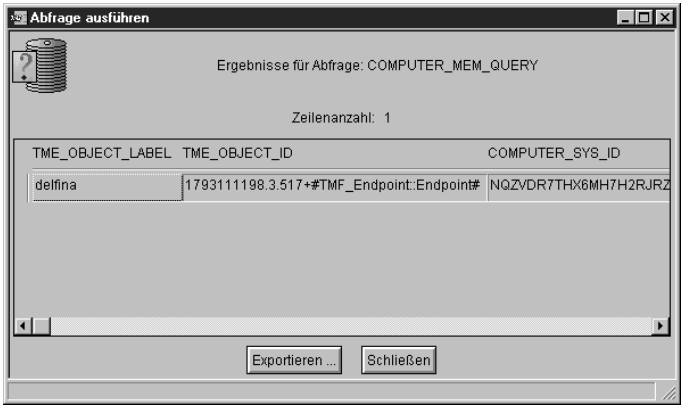
Sie können eine Abfrage entweder über die Tivoli-Arbeitsoberfläche oder die Befehlszeile ausführen.

Klicken Sie in einer Abfragebibliothek mit der rechten Maustaste auf das Symbol einer Abfrage, und wählen Sie die Option **Abfrage ausführen** aus.

-ODER-

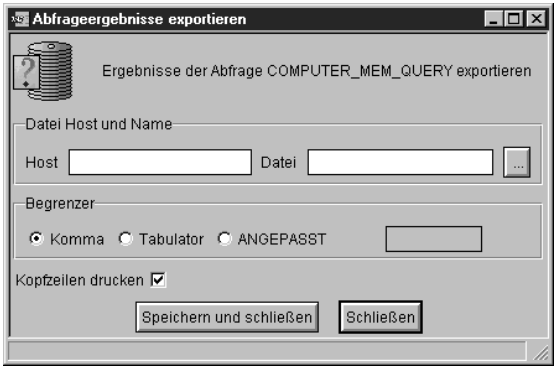
Wählen Sie im Dialog **Abfrage erstellen** bzw. **Abfrage editieren** die Option **Abfrage ausführen** aus.

Daraufhin wird der Dialog **Abfrage ausführen** geöffnet, der das Ergebnis der Abfrage in tabellarischer Form enthält.



Das Ergebnis der Abfrage können Sie über den Dialog **Abfrage ausführen** sichern. Gehen Sie hierzu wie folgt vor:

1. Klicken Sie im Dialog **Abfrage ausführen** auf **Exportieren**. Daraufhin wird der Dialog **Abfrageergebnisse exportieren** geöffnet.



2. Geben Sie im Feld **Host** den Namen des verwalteten Knotens ein, auf dem die Ergebnisdatei gespeichert werden soll. Erfolgt keine Angabe, wird eine Fehlernachricht ausgegeben.

3. Geben Sie im Feld **Datei** ein Verzeichnis und einen Namen für die Datei mit den Abfrageergebnissen an.

   -ODER-

   Klicken Sie auf die Schaltfläche mit den Auslassungszeichen (**...**), um das Dateisystem zu durchsuchen, und wählen Sie ein Verzeichnis und einen Dateinamen aus.

4. Wählen Sie eine der Optionen unter **Begrenzer** aus, um anzugeben, welches Trennzeichen zwischen den einzelnen Einträgen in der Datei mit den Abfrageergebnissen verwendet werden soll. Wenn ein anderer Begrenzer als Komma (,) oder Tabulator verwendet werden soll, wählen Sie **ANGEPASST** aus, und geben Sie im Feld rechts daneben den gewünschten Begrenzer ein.

5. Soll die Ausgabedatei den Namen der Abfrage sowie die Namen der Spalten enthalten, aktivieren Sie das Kontrollkästchen **Kopfzeilen drucken**.

6. Klicken Sie auf die Schaltfläche **Speichern und schließen**; die Datei wird daraufhin erstellt.

Über die Befehlszeile

Sie können mit dem Befehl **wrunquery** Abfragen ausführen und die Ergebnisse entweder an der standardmäßigen Ausgabeeinheit anzeigen oder in einer Datei sichern. Um die Abfrage **Win-machines** auszuführen und das Ergebnis in einer Datei zu speichern, müssen Sie den Befehl **wrunquery** wie folgt eingeben:

```
wrunquery -n -h amon -f /tmp/query.txt -d ";" Win-machines
```

Dabei gilt:

-n Gibt an, dass in die Ergebnisdatei keine Kopfzeilen übernommen werden sollen.

-h amon
Gibt an, dass die Datei auf dem verwalteten Knoten **amon** gespeichert werden soll.

-f /tmp/query.txt
Gibt **/tmp/query.txt** als Verzeichnis und Name der Datei mit den Abfrageergebnissen an.

-d ";" Gibt das Semikolon als Begrenzer an.

Win-machines
Gibt den Namen der Abfrage an, die ausgeführt werden soll.

Im Handbuch *Tivoli Management Framework Reference Manual* finden Sie weitere Informationen zum Befehl **wrunquery**.

Subskribenten über Abfragen ermitteln

Mit Hilfe von Abfragen können Sie nicht nur Informationen zu Ihrer Tivoli-Umgebung abrufen, sondern außerdem Clients in Ihrer Umgebung ermitteln, die bestimmte Kriterien erfüllen. Beispielsweise können Sie über eine Abfrage die Ziele für die Verteilung von Inventory-Profilen auswählen. Bei Ausführung der Abfrage werden die Clients aus dem Richtlinienbereich zurückgegeben, die den Suchkriterien entsprechen.

In Tivoli Inventory stehen Ihnen mehrere Standardabfragen zur Verfügung, mit denen Sie die Subskribenten für ein Inventory-Profil ermitteln können. Diese Abfragen befinden sich in der Abfragebibliothek SUBSCRIPTION_QUERY. Informationen zur Installation von Subskriptionsabfragen können Sie dem Abschnitt „Tivoli Inventory- und Subskriptionsabfragen installieren” auf Seite 87 entnehmen. Eine Liste der verfügbaren Subskriptionsabfragen sowie eine entsprechende Beschreibung finden Sie im Abschnitt „Subskriptionsabfragen” auf Seite 436.

In der folgenden Tabelle sind die dafür erforderlichen Berechtigungsklassen aufgeführt:

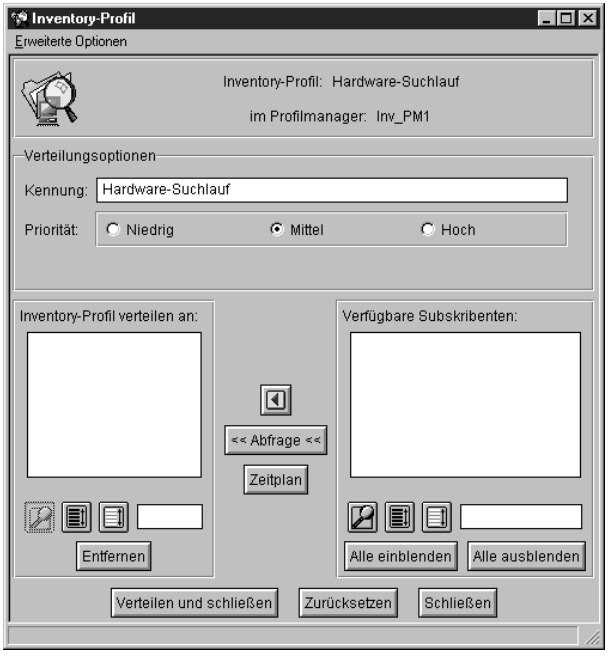
Operation	Kontext	Erforderliche Berechtigungsklasse
Ziele für die Verteilung eines Inventory-Profils auswählen	Inventory-Profil	**senior** oder **super**

Sie können die Ziele entweder über die Tivoli-Arbeitsoberfläche oder über die Befehlszeile auswählen.

Bei Ausführung einer Abfrage über den Dialog **Inventory-Profil** müssen Sie die Spalten TME_OBJECT_ID und TME_OBJECT_LABEL in die Liste **Ausgewählte Spalten** aufnehmen.

Folgende Schritte sind erforderlich, um über die Ausführung einer Abfrage im Dialog **Inventory-Profil** die Subskribenten für ein Profil zu definieren:

1. Wählen Sie aus dem Kontextmenü eines Inventory-Profils die Option **Verteilen** aus. Daraufhin wird der Dialog **Inventory-Profil** geöffnet.



2. Klicken Sie auf **Abfrage**. Daraufhin wird der Dialog **Abfrage ausführen** geöffnet.

3. Doppelklicken Sie in der Liste **Abfragebibliotheken** auf die Abfragebibliothek, die die gewünschte Abfrage enthält. Diese Liste enthält alle Abfragebibliotheken, die in dem Richtlinienbereich vorhanden sind. Bei Auswahl einer Abfragebibliothek werden die entsprechenden Abfragen in der Liste **Abfragen** angezeigt.
4. Wählen Sie in der Liste **Abfragen** eine Abfrage aus.
5. Klicken Sie auf **Ausführen**. Daraufhin wird der Dialog 'Abfrage ausführen' geschlossen und der Dialog 'Inventory-Profil' geöffnet. Bei der Abfrage werden die Subskribenten im Richtlinienbereich, die den Suchkriterien entsprechen, in die Liste **Inventory-Profil verteilen an** verschoben.

Befehlszeile

Der Befehl **wruninvquery** führt eine Abfrage aus und gibt die Ergebnisse in einem IDL-Format zurück, das Sie in Verbindung mit Skripts zum Ausfüllen von Subskriptionslisten verwenden können. Abfragen, die mit diesem Befehl ausgeführt werden, müssen die Spalten TME_OBJECT_ID und TME_OBJECT_LABEL enthalten. Der Befehl **wrunquery** dagegen kann für die Ausführung beliebiger Abfragen verwendet werden; das Ergebnis wird entweder an die standardmäßige Ausgabeeinheit geschickt oder in einer Datei gespeichert.

Geben Sie zum Ausführen der Abfrage **Win-machines** den folgenden Befehl ein:

```
wruninvquery -l Win-machines
```

Dabei gilt:

-l Gibt die Namen der Maschinen an, die den Suchkriterien entsprechen, nicht deren Objekt-ID.

Win-machines
Gibt an, dass die Abfrage **Win-machines** ausgeführt werden soll.

Weitere Informationen zu den Befehlen **wruninvquery** und **wrunquery** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Informationen zu einzelnen Clients abrufen

Sie können mit Abfragen nicht nur Informationen zur Tivoli-Umgebung ermitteln oder Listen mit potenziellen Zielen erstellen, sondern außerdem auch die in Inventory gespeicherten Informationen zu einzelnen Clients in der Tivoli-Umgebung abrufen. Von der Abfrage werden die Inventory-Suchergebnisse für die betreffende Maschine zurückgegeben. Beim Abrufen von Informationen zu einzelnen Clients müssen in der Abfrage die Spalten TME_OBJECT_ID und TME_OBJECT_LABEL angegeben werden.

In der folgenden Tabelle sind die Berechtigungsklassen aufgeführt, die für die Ausführung von Abfragen über den Dialog **Profilmanager** erforderlich sind:

Operation	**Kontext**	**Erforderliche Berechtigungsklasse**
Abfrage zu einem einzelnen Client ausführen	Der Richtlinienbereich, der die Abfragebibliothek der gültigen Abfrage enthält	**RIM_view**, **Query_view**, **Query_execute**, **Query_edit**, **senior**, oder **super**

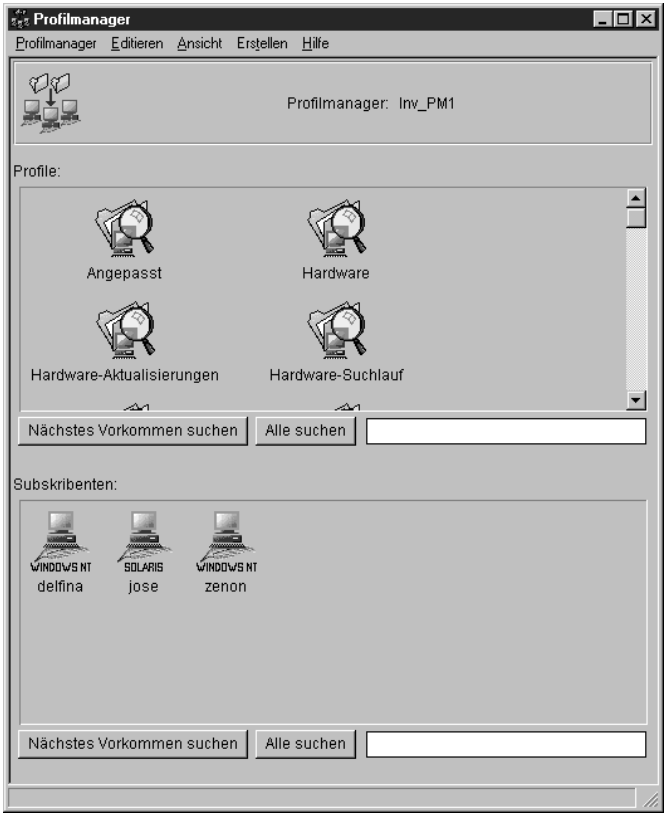
Sie können Inventory-Daten zu einem Client entweder über die Tivoli-Arbeitsoberfläche oder die Befehlszeile abrufen.

Über die Arbeitsoberfläche

Folgende Schritte sind zum Anzeigen von Informationen über das Symbolmenü eines Endpunkts erforderlich:

1. Klicken Sie im Dialog 'Profilmanager' mit der rechten Maustaste auf das Symbol des gewünschten Endpunkts.

   Anmerkung: Die Symbole von Endpunkten sind nur in dem Fenster des Profilmanagers enthalten, dem sie als Subskribenten zugeordnet sind.



2. Folgende Schritte sind zum Anzeigen von Informationen zum Endpunkt erforderlich:

   a. Wählen Sie im Kontextmenü die Option **Abfrage ausführen** aus. Daraufhin wird der Dialog **Abfrage ausführen** geöffnet.

   b. Doppelklicken Sie in der Liste **Abfragebibliotheken** auf die Abfragebibliothek, die die gewünschte Abfrage enthält. Diese Liste enthält alle Abfragebibliotheken, die in dem Richtlinienbereich vorhanden sind. Bei Auswahl einer Abfragebibliothek werden die entsprechenden Abfragen in der Liste **Abfragen** angezeigt.

   c. Wählen Sie in der Liste **Abfragen** eine Abfrage aus.

   d. Klicken Sie auf **Ausführen**. Daraufhin wird die Abfrage von Inventory ausgeführt und nur das Ergebnis für die ausgewählte Maschine angezeigt.

Über die Befehlszeile

Mit dem Befehl **wqueryinv** können Sie Inventory-Daten zu einem bestimmten Client abrufen.

```
wqueryinv -n -d ";" -h amon -f /tmp/x1.txt \
-q PARTITION_QUERY -m tacoma
```

Dabei gilt:

–n Gibt an, dass die Datei mit dem Abfrageergebnis keine Kopfzeilen enthalten soll.

–d “;” Gibt an, dass die einzelnen Einträge in der Datei mit dem Abfrageergebnis durch Semikolons voneinander getrennt werden.

–h amon
Gibt den Namen des Hosts an, auf dem die Datei mit dem Abfrageergebnis gespeichert wird.

–f /tmp/x1.txt
Gibt das Verzeichnis und den Namen der Datei mit dem Abfrageergebnis an.

–q PARTITION_QUERY
Gibt den Namen der Abfrage an, die ausgeführt werden soll.

–m tacoma
Gibt die Tivoli-Objekt-ID der Maschine an, zu der Inventory-Daten abgerufen werden sollen, und gibt außerdem an, dass es sich bei der Maschine um einen verwalteten Knoten handelt.

Weitere Informationen zum Befehl **wqueryinv** können Sie dem Abschnitt „Befehle” auf Seite 239 entnehmen.

Abfragen im TMR-Verbund verwenden

In diesem Abschnitt wird erläutert, wie Sie Abfragen für miteinander verbundene TMRs erstellen und Daten jeder einzelnen TMR anzeigen können. Manche Benutzer von Tivoli Inventory beispielsweise erfassen Daten für eine Vielzahl interner oder externer Kunden. Befinden sich die Systeme der einzelnen Kunden in verschiedenen TMRs, können Sie mit folgender Vorgehensweise die Daten jedes einzelnen Kunden anzeigen.

Zum Anzeigen von Daten jeder einzelnen TMR erstellen Sie eine Abfrage, die die verschiedenen Daten mit Hilfe der Objekt-ID (OID) unterscheidet. Eine OID ist eine dreiteilige Kennung mit dem Format *n.n.n*, zum Beispiel 1264987995.1.326. Die OID setzt sich aus den folgenden Werten zusammen:

- Der Nummer der Region. Die Nummer der Region von Objekten in Tivoli Management Framework Version 3.0 und höher ist zehnstellig (wie die Nummer 1264987995 im oben genannten Beispiel). Sie wird bei der Installation des TMR-Servers generiert. Für jede TMR wird mit Hilfe eines Algorithmus eine Region-Nummer generiert. In den meisten Fällen wird die generierte Nummer nur einmal vergeben. Durch das Zufallselement bei der Generierung der Nummern ist jedoch nicht gänzlich auszuschließen, dass zwei TMRs ein und dieselbe Region-Nummer generieren. In diesem Fall ist eine der Regionen erneut zu installieren, damit die Verbindung zu der Region mit derselben Nummer hergestellt werden kann.
- Der Nummer des Objekt-Dispatchers (oserv). Diese Nummer wird gelegentlich auch als Host-Nummer bezeichnet und in der Installationsreihenfolge der verwalteten Knoten vergeben. Der TMR-Server ist Dispatcher 1, der nächste installierte verwaltete Knoten Dispatcher 2, und so weiter.
- Der Nummer des Objekts. Dabei handelt es sich um eine Nummer für das Objekt selbst.

Wenn ein verwalteter Knoten oder Endpunkt in einer TMR erstellt wird, stimmt die Nummer der Region des verwalteten Knotens bzw. des Endpunkts mit der des TMR-Servers überein.

Wenn Sie feststellen möchten, welche Systeme in einer bestimmten TMR vorhanden sind, editieren Sie die Abfrage INVENTORY_HWARE, um ausschließlich die Spalten TME_OBJECT_LABEL, TME_OBJECT_ID und COMPUTER_SYS_ID auszuwählen. Geben Sie die folgende Klausel im Textfenster 'Zusätzliche Klauseln' ein:

```
Reihenfolge durch TME_OBJECT_ID
```

Wenn Sie diese Abfrage ausführen, wird jede in den TMRs vorhandene Endpunktgruppierung zurückgegeben.

Wenn Sie bestimmte Daten für eine TMR abfragen möchten, editieren Sie die gewünschte Abfrage, indem Sie die folgende WHERE-Klausel hinzufügen:

```
TME_OBJECT_ID like 'nnnnnnnnn%'
```

Dabei steht *nnnnnnnnnn* für die ersten 10 Ziffern der OID. Wenn Sie Daten in mehreren TMRs abfragen möchten, fügen Sie für jede OID eine WHERE-Klausel hinzu.

Die für ein System installierten Hardware- und Softwareinformationen sind an eine Computersystem-ID gebunden. Wenn Sie Informationen über einen bestimmten Endpunkt anzeigen möchten, fügen Sie die folgende WHERE-Klausel hinzu:

```
COMPUTER_SYS_ID = 'computer_sys_id'
```

Dabei steht *computer_sys_id* für den Wert der COMPUTER_SYS_ID des gewünschten Endpunkts.

A

Berechtigungsklassen

Dieser Anhang enthält Tabellen, in denen die verschiedenen Aktivitäten und Berechtigungsklassen erläutert werden, die für die Ausführung von Aktivitäten mit Hilfe von Tivoli Inventory erforderlich sind.

Inventory-Berechtigungsklassen

Für die Arbeit mit Tivoli Inventory müssen Sie als Tivoli-Administrator angemeldet sein. Je nach Aktivitäten, die ausgeführt werden sollen, müssen Sie über eine oder mehrere der folgenden Berechtigungsklassen verfügen, und zwar entweder in der TMR (Tivoli Management Region) oder in dem Richtlinienbereich, in dem die betreffende Operation ausgeführt wird:

- **Inventory_end_user**
- **Inventory_view**
- **Inventory_query**
- **Inventory_scan**
- **Query_view**
- **Query_edit**
- **Query_execute**
- **RIM_view**
- **RIM_update**

- **user**
- **admin**
- **senior**
- **super**
- **Install_product**
- **Install_client**
- **restore**
- **backup**

Inventory-Operationen

In der folgenden Tabelle werden die Berechtigungsklassen aufgeführt, die für die Ausführung von Tivoli Inventory-spezifischen Operationen erforderlich sind:

Operation	Erforderliche Berechtigungsklasse
Tivoli Inventory installieren	**Install_product** oder **super**
Webfunktionen von UserLink for Tivoli Inventory verwenden	**Inventory_end_user**
Inventory-Profil erstellen	**senior** oder **super**
Inventory-Profil anpassen oder editieren	**admin**, **senior** oder **super**
Inventory-Profil anzeigen	**Inventory_view**, **user**, **admin**, **senior** oder **super**
Inventory-Profil klonen	**senior** oder **super**
Inventory-Profil löschen	**senior** oder **super**
Subskribenten von Inventory-Profilen festlegen	**admin**, **senior** oder **super**
Inventory-Profil verteilen	**Inventory_scan** oder **admin**
Inventory-Daten im Konfigurations-Repository anzeigen	**Query_view**, **RIM_view**, **user**, **admin**, **senior** oder **super**
Webseiten für Tivoli Inventory-Administratoren verwenden	**RIM_view**

Operation	Erforderliche Berechtigungsklasse
Softwaresignaturen hinzufügen, löschen bzw. editieren	**RIM_update**, **senior** oder **super**

Abfrageoperationen

In der folgenden Tabelle werden die Berechtigungsklassen aufgeführt, die für die Ausführung von abfragespezifischen Operationen erforderlich sind:

Operation	Erforderliche Berechtigungsklasse
Abfragebibliothek erstellen	**senior** oder **super**
Abfrage erstellen	**admin**, **senior** oder **super**
Abfrage editieren	**Query_edit**, **admin**, **senior** oder **super**
Abfrage anzeigen	**Query_execute, Query_edit, Query_view**, **user**, **admin**, **senior** oder **super**
Abfrage ausführen	**Query_edit, Query_execute**, **admin**, **senior** oder **super**

B

Befehle

In diesem Anhang sind in alphabetischer Reihenfolge alle Tivoli-Befehle aufgeführt und erläutert, die für Tivoli Inventory relevant sind. Informationen zu den für die Tivoli-Abfragefunktion und das RIM (Relational Database Management System (RDBMS) Interface Module) relevanten Befehlen finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.

Die meisten Tivoli-Befehle beginnen mit einem **w**, und Vokale werden oft ausgelassen, um den Befehl so kurz wie möglich zu halten. Die Befehlsnamen basieren normalerweise auf dem Format **w+Verb+Objekt**. Beispielsweise werden die allgemeinen Merkmale eines Inventory-Profils mit dem Befehl **wsetinvglobal** festgelegt. Und angezeigt werden die allgemeinen Merkmale mit dem Befehl **wgetinvglobal**.

Tivoli-Befehle verwenden

In vielen Fällen ist die Ausführung Tivoli-spezifischer Operationen über die Befehlszeile bequemer als über die Arbeitsoberfläche, oft ist sie auch erforderlich. Im Folgenden einige Situationen, in denen die Arbeit mit der Befehlszeilenschnittstelle (CLI = Command Line Interface) empfohlen wird:

- Sie haben keinen Zugriff auf eine Tivoli-Arbeitsoberfläche.
- Mehrere Tivoli-Befehle sollen in einem Shell-Skript bzw. einer Stapeldatei zusammengefasst werden.

- Sie möchten einen Befehl im Rahmen eines Skripts ausführen, die mehrere Operationen ausführt.

Befehlszeilensyntax

Auf den folgenden Seiten in diesem Anhang werden bei der Erläuterung der Befehlssyntax die folgenden Sonderzeichen verwendet:

[ ] Gibt optionale Argumente an. Optionen, die nicht in eckigen Klammern stehen, sind erforderlich.

... Gibt an, dass für die vorangehende Option mehrere Werte angegeben werden können.

| Gibt an, dass sich Argumente gegenseitig ausschließen. Sie können entweder die Option rechts oder die Option links neben dem Trennzeichen angeben. Die Angabe beider Optionen innerhalb eines Befehls ist nicht zulässig.

{ } Optionen, die sich gegenseitig ausschließen und von denen eine erforderlich ist, werden in geschweifte Klammern gesetzt. Optionale Angaben werden hingegen in eckige Klammern gesetzt ([ ]).

Beispiel

wruninvquery [**–i**] [**–T** *IDL-Art*] [**–l** | **–t**] *Abfrage* [*Eingabe*]...

Das Auslassungszeichen (...) hinter der Option *Eingabe* gibt an, dass mehrere Abfragen angegeben werden können. Wenn die Optionen **–l** oder **–t** angegeben werden sollen, können Sie in jedem Befehl jeweils nur eine dieser Optionen verwenden. Die beiden Optionen sind durch einen senkrechten Strich | (logisches Oder-Zeichen) getrennt, d. h. sie schließen sich gegenseitig aus. Die Option *Abfrage* ist erforderlich, alle anderen Angaben sind optional, was durch die eckigen Klammern ([ ]) erkennbar ist.

Objektverweise

Bei Objektverweisen in einem Befehl handelt es sich nicht um absolute Objektverweise, wie sie in der Programmierung verwendet werden. Stattdessen werden benutzerfreundliche Namen verwendet, die sich von dem Objektnamen ableiten, der von dem Benutzer der betreffenden Anwendung vergeben wurde, z. B. bei der Erstellung eines Richtlinienbereiches.

In CLI-Befehlen können zwei verschiedene Namensformate verwendet werden:

- Registrierte Namen
- Objektpfade

Tivoli-Befehle unterstützen beide Namensformate. In manchen Fällen kann sich ein Namensformat unter Umständen als vorteilhafter erweisen als das andere. Meldet beispielsweise eine Fehlernachricht, dass eine bestimmte Ressource nicht gefunden wurde, können Sie den jeweiligen Vorgang unter Verwendung einer anderen Namenskonvention wiederholen.

Registrierte Namen

Die Tivoli-Namensregistrierdatenbank enthält registrierte Namen. Bei einem *registrierten Namen* handelt es sich um den Namen, unter dem eine Ressourceninstanz bei ihrer Erstellung in der Namensregistrierdatenbank registriert wird. Es gibt verschiedene Ressourcenarten, und jeder Ressource ist ein Name zugewiesen. Bei dem Drucker 'lp01' beispielsweise handelt es sich um die Ressource mit dem Namen 'lp01', die Ressourcenart ist 'Drucker'. Beispiele für die Verwendung von registrierten Namen als Optionen in den Befehlen **wls** und **wmv**:

```
wls @PolicyRegion:Servers
wmv @ManagedNode:ayers-rock @PolicyRegion:Servers
```

Die Syntax für die Angabe einer Ressource mit Hilfe des registrierten Namens ist *Art:Name*; dabei steht *Art* für die Ressourcenart und *Name* für die Ressourceninstanz, für die eine Operation ausgeführt werden soll.

In der Namensregistrierdatenbank darf zwei Ressourcen derselben Art innerhalb einer TMR (Tivoli Management Region) nicht derselbe Name zugewiesen werden. Die Verwendung desselben Ressourcennamens in zwei oder mehreren miteinander verbundenen TMRs ist dagegen zulässig. Bei dem Versuch, eine Operation für eine Ressource auszuführen, deren Name mehr als einmal innerhalb einer TMR vergeben wurde, wird eine Fehlernachricht zurückgegeben; der Vorgang wird nicht ausgeführt. Solche Fälle lassen sich vermeiden, indem Sie eine der Ressourcen umbenennen oder den Namen des Bereichs an den Ressourcennamen anhängen, um so eine Unterscheidung zu treffen, wie im folgenden Beispiel:

```
wls @ManagedNode:moria#moria-Region
```

Weitere Informationen zu den Befehlen **wls** und **wmv** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.

Objektpfade

Objektpfade sind mit Pfadnamen in Dateisystemen vergleichbar; sie können relativ oder absolut sein. Ein absoluter Pfad beginnt mit einem Schrägstrich (/). Ein relativer Pfad kann mit einem beliebigen Zeichen beginnen, einschließlich der besonderen Pfadkomponenten '.' und '..'. Beispiele für die Verwendung von Objektpfaden als Optionen in den Befehlen **wls** und **wmv**:

```
wls /Regions/Servers
wmv../Servers/ayers-rock /Regions/Servers
```

Die Syntax für die Angabe einer Ressource mit Hilfe des Objektpfades als Name lautet /*Ressourcenart*/*Ausgangsverzeichnis*/[*Art*:]*Name*; dabei steht *Ressourcenart* für die Ressourcenart, *Ausgangsverzeichnis* für den Beginn des Objektpfades, *Art* für weitere Angaben zu der Ressource und *Name* für die entsprechende Instanz, für die eine Operation ausgeführt werden soll. In vielen Fällen wird das optionale Qualifikationsmerkmal *Art* angegeben, wenn eine weitere Ressource mit demselben Namen vorhanden ist, bei der es sich jedoch um eine andere Ressourcenart handelt.

Beispiele

Wenn ein Befehl zu lang für die in diesem Dokument verwendeten Zeilen ist, wird die Zeile durch einen umgekehrten Schrägstrich (\) umgebrochen. Bei der tatsächlichen Eingabe des Befehls darf die Zeile jedoch nicht umgebrochen werden.

Befehle für Tivoli Inventory

In Tivoli Inventory steht eine Befehlszeilenschnittstelle (CLI) zur Verfügung, über die viele der Operationen ausgeführt werden können, deren Ausführung auch über die Arbeitsoberfläche möglich ist. In Tivoli Inventory sind folgende Befehle verfügbar:

Befehl	Funktion
wdistinv	Mit diesem Befehl werden Inventory-Profile verteilt.
wepscan	Mit diesem Befehl werden Suchläufe auf einem Endpunkt eingeleitet.
wgetinvglobal	Mit diesem Befehl werden die für ein Inventory-Profil festgelegten allgemeinen Merkmale aufgelistet.
wgetinvpcfiles	Mit diesem Befehl werden Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, aufgelistet.
wgetinvpchw	Mit diesem Befehl werden die für einen Hardwaresuchlauf auf einem PC-Endpunkt festgelegten Optionen aufgelistet.
wgetinvpcsw	Mit diesem Befehl werden die Optionen für PC-Softwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden.
wgetinvunixfiles	Mit diesem Befehl werden Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, aufgelistet.

Befehl	Funktion
wgetinvunixhw	Mit diesem Befehl werden die für einen Hardwaresuchlauf auf einem UNIX-Endpunkt festgelegten Optionen aufgelistet.
wgetinvunixsw	Mit diesem Befehl werden die Optionen für UNIX-Softwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden.
wcrtinvcb	Mit diesem Befehl wird das Callback-Objekt von Inventory erstellt.
winvfilter	Mit diesem Befehl werden die Dateien in der Liste der angepassten Filter, die im Konfigurations-Repository gespeichert ist, geändert.
winvrmnode	Mit diesem Befehl werden alle durch Suchläufe erfassten Informationen zu einem bestimmten System aus dem Konfigurations-Repository entfernt.
winvsig	Mit diesem Befehl wird die Liste der Softwaresignaturen im Konfigurations-Repository geändert.
wqueryinv	Mit diesem Befehl werden im Konfigurations-Repository Informationen zu einem Endpunkt abgefragt.
wsetinvglobal	Mit diesem Befehl werden die allgemeinen Merkmale für ein Inventory-Profil festgelegt.
wsetinvpcfiles	Mit diesem Befehl werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, angegeben.
wsetinvpchw	Mit diesem Befehl werden die Optionen für Hardwaresuchläufe auf PC-Endpunkten festgelegt.
wsetinvpcsw	Mit diesem Befehl werden die Optionen für Softwaresuchläufe auf PC-Endpunkten festgelegt.
wsetinvunixfiles	Mit diesem Befehl werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, angegeben.

Befehl	Funktion
wsetinvunixhw	Mit diesem Befehl werden die Optionen für Hardwaresuchläufe auf UNIX-Endpunkten festgelegt.
wsetinvunixsw	Mit diesem Befehl werden die Optionen für Softwaresuchläufe auf UNIX-Endpunkten festgelegt.

Befehle für SCS

In der folgenden Tabelle werden die SCS (Scalable Collection Service)-Befehle, die Sie in Verbindung mit Tivoli Inventory verwenden können, beschrieben. Derzeit können SCS-Tasks nur über die Befehlszeilenschnittstelle (CLI) ausgeführt werden.

Befehl	Funktion
wcollect	Mit diesem Befehl werden aktuelle Konfigurationsdaten eines Kollektors zurückgegeben, Kollektoren gestartet und gestoppt, die im Kollektor zwischengespeicherten Kollektor-Routes zurückgesetzt und Optionen für einen Kollektor konfiguriert.
wcrtinvcb	Mit diesem Befehl wird eine Instanz des Callback-Objekts von Inventory erstellt. (In jeder TMR kann nur eine Instanz erstellt werden.)
wcrtinvdh	Mit diesem Befehl wird eine Instanz des Inventory Data Handler-Objekts erstellt. (In jeder TMR kann nur eine Instanz erstellt werden.)
wcstat	Mit diesem Befehl werden Statusinformationen zu einem Kollektor, Informationen zu einer CTOC (collection table of contents) sowie der Inhalt der Warteschlangen eines Kollektors zurückgegeben.
wgetinvdh	Mit diesem Befehl werden Konfigurationsdaten des Inventory Data Handler zurückgegeben.
wgetscanstat	Mit diesem Befehl werden Informationen über die aktuellen Inventory-Suchläufe zurückgegeben.
wsetinvdh	Mit diesem Befehl werden die Einstellungen für den Inventory Data Handler geändert.

Diese Befehl können in zwei Hauptgruppen unterschieden werden:

- Befehle für SCS-Komponenten

  Mit diesem Befehl können Sie Informationen über Kollektoren konfigurieren und abrufen. Dazu gehören die Befehle **wcollect** und **wcstat**. (Mit dem Befehl **wcollect** können Sie darüber hinaus einige Optionen für den Inventory Data Handler konfigurieren.)

- Befehl für Tivoli Inventory-Komponenten

  Mit Hilfe dieser Befehle können Sie steuern, wie Tivoli Inventory mit SCS zusammenarbeitet, und auf SCS bezogene Informationen von Tivoli Inventory-Komponenten abrufen. Diese Befehle können in zwei Kategorien eingeordnet werden:

  - Befehle, die in Verbindung mit Inventory-Profilen verwendet werden. Dazu gehören die Befehle **wgetinvglobal** und **wsetinvglobal**.
  - Befehle, die mit dem Inventory Data Handler und dem Statuskollektor verwendet werden. Dazu gehören die Befehle **wcrtinvdh**, **wgetinvdh**, **wgetscanstat** und **wsetinvdh**.

Befehlssyntax

Dieser Abschnitt enthält eine Beschreibung der Tivoli Inventory-Befehle einschließlich ihrer Syntax und Funktionen. Eine Übersicht über diese Befehle erhalten Sie auch, wenn Sie auf UNIX-Systemen den Befehl **man** eingeben.

wcollect

Mit diesem Befehl werden Konfigurationsdaten eines Kollektors zurückgegeben, Kollektoren gestartet und gestoppt, die im Kollektor zwischengespeicherten Kollektor-Routes zurückgesetzt und Optionen für einen Kollektor konfiguriert.

Syntax

wcollect [*Optionen*] *Kollektor*

Dabei wird *Kollektor* in einem der folgenden Formate angegeben: **@ManagedNode:***Kollektorname*, **@Gateway:***Kollektorname* oder **@InvDataHandler:inv_data_handler** (*Kollektorname* steht für den Namen des verwalteten Knotens oder Gateways). Die *Optionen* können aus einer der folgenden Kategorien ausgewählt werden:

- Optionen für aktive Kollektoren

  [**–s** | **–h immediate** | **graceful**]

- Optionen für Kollektorkonfiguration und -optimierung

  [**–d 0** | **1** | **2** | **3**]

  [**–g** *Größe des Debug-Protokolls*]

  [**–l** *Laufzeitposition*]

  [**–z** *Depotgröße*]

  [**–c** *Depotblockgröße*]

  [**–i** *Thread-Leerlaufzeit*]

  [**–p** *Thread-Inaktivierungszeit*]

  [**–t** *max. Zahl an Eingabe-Threads*]

  [**–m** *max. Zahl an Eingabe-Wiederholungen*]

  [**–o** *max. Zahl an Ausgabe-Threads*]

  [**–e** *Wiederholungsintervall*]

  [**–r**]

  [**–x** *Ausschlussbereich für Offlinks in Kollektor*]

  [**–f true** | **false**]

Beschreibung

Mit dem Befehl **wcollect** können drei wesentliche Funktionen ausgeführt werden. Zum einen können Sie Konfigurationsdaten eines Kollektors abrufen, indem Sie bei der Ausführung des Befehls den Kollektor angeben, dessen Daten Sie benötigen.

Zweitens können Sie mit dem Befehl **wcollect** einen vorhandenen Kollektor stoppen oder starten und im Kollektor zwischengespeicherte Route-Informationen löschen. Die Optionen, mit denen Sie diese Operationen ausführen können, werden Optionen für aktive Kollektoren genannt.

Drittens können Sie mit dem Befehl **wcollect** Kollektorattribute konfigurieren. Die Optionen, mit denen Sie diese Operationen ausführen können, werden Optionen für Kollektorkonfiguration und -optimierung genannt.

Optionen

Die Optionen können in solche für aktive Kollektoren und solche für Kollektorkonfiguration eingeteilt werden.

Aktive Kollektoren

–s Startet den Kollektor, nachdem er angehalten wurde.

–h *Option*

Hält den Kollektor an. Sie können eine der folgenden Optionen verwenden:

- **graceful** - Der Kollektor wird erst gestoppt, nachdem alle noch ausstehenden aktiven Datensammlungsvorgänge beendet wurden.
- **immediate** - Der Kollektor wird gestoppt, ohne das Ende der Verarbeitung von aktiven Datensammlungsvorgängen abzuwarten.

Kollektorkonfiguration

–d *Option*

Gibt die Protokollstufe für das Speichern von Debug-Informationen in der SCS-Protokolldatei an. Sie können eine der folgenden Optionen verwenden:

- **0** - Keine Protokollierung.
- **1** - nur Protokollierung schwer wiegender Fehler.
- **2** - Protokollierung von schwer wiegenden Fehlern und Warnungen.
- **3** - Protokollierung aller Debug-Nachrichten.

Der Standardwert ist 1. Die SCS-Protokolldatei 'mcollect-.log' befindet sich im Verzeichnis '$DBDIR/mcollect' des jeweiligen Kollektors.

–g *Größe des Debug-Protokolls*

Gibt die maximale Größe der Kollektorprotokolldatei 'mcollect.log' in Megabyte (MB) an. Bei Erreichen der maximalen Größe werden 90 Prozent des Dateiinhalts gelöscht, und nur die aktuellsten 10 Prozent bleiben erhalten. Standardmäßig ist als maximale Größe 1 MB angegeben.

–l *Laufzeitposition*

Gibt die Position des Laufzeitverzeichnisses für den Kollektor an. Das Laufzeitverzeichnis enthält die Depot- und Laufzeitdateien (Erweiterungen '*.dat' und '*.log'). Dieses Verzeichnis muss auf einem stabil laufenden Datenträger gespeichert werden, dessen freier Speicherbereich ausreicht, um eine dauerhafte Speicherung von Datensammlungen zu gewährleisten. Beim Laufzeitverzeichnis sollte es sich nicht um ein temporäres Verzeichnis handeln. Darüber hinaus müssen dem Benutzereintrag **tmersrvd** oder **nobody** (der Tivoli-Benutzereintrag ohne Berechtigungen) Berechtigungen für den Schreib- und Lesezugriff auf dieses Verzeichnis zugeordnet werden. Standardmäßig befindet sich das Laufzeitverzeichnis im Verzeichnis '$DBDIR/mcollect' und das Depot im Verzeichnis '$DBDIR/mcollect/depot' des jeweiligen Kollektors.

Anmerkung: Wenn Sie ein neues Laufzeitverzeichnis angegeben haben, muss der Kollektor gestoppt, und das Depot und alle sonstigen Daten müssen aus dem alten Laufzeitverzeichnis in das neue Verzeichnis verschoben werden.

–z *Depotgröße*

Gibt die Größe des Depotverzeichnisses in MB an. Standardmäßig ist als Größe 40 MB angegeben. Sie können die Depotgröße zwar erhöhen, können danach jedoch keinen kleineren Wert mehr angeben. Sie können die Depotgröße beispielsweise auf 50 MB erhöhen. Danach können Sie die Größe nochmals auf 55 MB erhöhen, doch Sie können Sie nicht mehr auf 45 MB verringern.

–c *Depotblockgröße*

Gibt die Größe des Übertragungsblocks in Kilobyte (KB) an. Wenn ein untergeordneter Kollektor Daten an den nächsten Kollektor sendet, werden die Daten in mehrere Einheiten, so genannte Übertragungsblöcke, aufgeteilt. Standardmäßig wird die Blockgröße mit 1024 KB angegeben.

Dieser Wert sollte so groß gewählt werden, dass die Datenübertragung effizient ausgeführt werden kann, jedoch auch so klein, dass die verfügbare Bandbreite möglichst effizient ausgenutzt werden kann.

–i *Thread-Leerlaufzeit*

Gibt an, wie viele Sekunden lang ein Thread sich im Leerlauf befinden kann, bevor er abgeschaltet wird. Der Standardwert für die Leerlaufzeit ist 60 Sekunden.

–p *Thread-Inaktivierungszeit*

Gibt an, wie viele Sekunden lang ein Thread *inaktiviert* sein (warten) kann, wenn System- oder Netzwerkeinschränkungen erreicht wurden. Der Thread versucht, den Datensammlungsprozess wieder aufzunehmen, nachdem die Inaktivierungszeit abgelaufen ist. Der Standardwert für die Inaktivierungszeit ist 5 Sekunden.

–t *max. Anzahl Eingabe-Threads*

Gibt die maximale Anzahl der Eingabe-Threads an, die der Kollektor gleichzeitig verarbeiten kann. Dieser Wert sollte so groß gewählt werden, dass die maximale Kollektorleistung genutzt werden kann, jedoch auch so klein, dass nicht zu viele Systemressourcen belegt werden. Der Standardwert ist 5.

–m *max. Anzahl Eingabewiederholungen*

Gibt die maximale Anzahl Wiederholungen für die Verarbeitung einer Datensammlungsanforderung von einem untergeordneten Kollektor an. . Der Standardwert ist 10. Mit Hilfe der Option **–e** *Wiederholungsintervall* können Sie festlegen, nach wie vielen Sekunden ein weiterer Wiederholungsversuch gestartet werden soll.

–o *max. Anzahl an Ausgabe-Threads*

Gibt die maximale Anzahl der Ausgabe-Threads an, die der Kollektor gleichzeitig verarbeiten kann. Der Standardwert ist 5. Dieser Wert sollte so groß gewählt werden, dass die maximale Kollektorleistung genutzt werden kann, jedoch auch so klein, dass nicht zu viele Systemressourcen belegt werden. Es ist zu empfehlen, für diese Option denselben Wert festzulegen wie für die Gesamtanzahl von RDBMS-Verbindungen aller RIM-Objekte, die von dem Inventory Data Handler verwendet werden. Weitere Informationen zum Festlegen von RDBMS-Verbindungen für RIM-Objekte finden Sie unter „RIM-Objekte konfigurieren” auf Seite 107.

–e *Wiederholungsintervall*

Legt fest, nach wie vielen Sekunden ein weiterer Versuch zur Verarbeitung einer Eingabe- oder Ausgabeanforderung an den Kollektor gestartet werden soll. Der Standardwert ist 1 Sekunde. Mit Hilfe der Option **–m** *max. Anzahl Eingabewiederholungen* können Sie die maximale Anzahl der Wiederholungsversuche angeben.

–r Entfernt lokal zwischengespeicherte Kollektor-Routes des jeweiligen Kollektors. Wenn der Kollektor nach der Ausführung dieses Befehls die nächste Datensammlungsanforderung verarbeitet, lädt er neue Route-Informationen vom Kollektor-Manager.

Wenn Sie mit Hilfe des Befehls **wrpt** Änderungen in der Kollektorhierarchie vorgenommen haben, sollten Sie anschließend diese Option für alle betroffenen Kollektoren ausführen. Dadurch werden die auf diesen Kollektoren gespeicherten Route-Informationen synchronisiert.

–x *Ausschlussbereich für Offlinks in Kollektor*
Gibt den Ausschlussbereich für Offlinks an. Offlinks sind die Objekt-Dispatcher-Nummern der Kollektoren, von denen der angegebene Kollektor keine Daten sammeln darf. Das heißt, Sie können die Verbindungen zu den Kollektoren abschalten, deren Objekt-Dispatcher-Nummer Sie für diese Option angeben.

Sie können die Objekt-Dispatcher-Nummern einzeln in aufsteigender numerischer Reihenfolge, jeweils getrennt durch Kommas, wie im folgenden Beispiel angeben:

```
wcollect -x "4,5,6,7" Kollektor
```

Oder Sie können durch einen Bindestrich zwischen den Zahlen einen ganzen Bereich von Objekt-Dispatcher-Nummern angeben:

```
wcollect -x "4-7" Kollektor
```

Durch die Einstellung dieser Optionen können Sie den Datenaustausch im Netzwerk zwischen untergeordneten Kollektoren steuern. Beispielsweise können Sie bei starkem Datenaustausch im Netzwerk die Verbindungen zu einigen Kollektoren abschalten und erst wieder herstellen, wenn der Datenaustausch im Netzwerk geringer ist. Diese Option hat keinen Einfluss auf die Datenerfassung zwischen Endpunkten und dem ersten Gateway-Kollektor.

Wenn Sie alle Verbindungen, die Sie zuvor abgeschaltet haben, gleichzeitig wiederherstellen möchten, geben Sie für diese Option eine leere Zeichenfolge (″ ″) als Wert an.

Standardmäßig sind keine Objekt-Dispatcher-Nummern angegeben.

–f *Option*
Gibt an, ob Informationen zu beendeten CTOCs in die Protokolldatei geschrieben werden sollen. Auf jeder Stufe der Datensammlung können CTOCs mit Hilfe des Befehls **wcstat** protokolliert werden. Sie können eine der folgenden Optionen verwenden:

true Informationen zu beendeten CTOCs werden in die Protokolldatei geschrieben. Dies ist die Standardoption.

false Informationen zu beendeten CTOCs werden nicht in die Protokolldatei geschrieben.

@Gateway:*Kollektorname*
Gibt den Namen des Kollektors an, auf dem dieser Befehl ausgeführt werden soll. Das Format **@Gateway:***Kollektorname* wird für Kollektoren auf Gateways verwendet.

@ManagedNode:*Kollektorname*
Gibt den Namen des Kollektors an, auf dem dieser Befehl ausgeführt werden soll. Das Format **@ManagedNode:***Kollektorname* wird für Kollektoren auf verwalteten Knoten verwendet, die nicht als Gateway konfiguriert wurden.

@InvDataHandler:inv_data_handler
Gibt an, dass dieser Befehl mit dem Inventory Data Handler ausgeführt werden soll.

Autorisierung

senior

Hinweise

Sie können Optionen für die Kollektorkonfiguration jederzeit ausführen, müssen anschließend jedoch die betroffenen Kollektoren erneut starten, damit die Änderungen übernommen werden. Kollektoren können mit Hilfe der Optionen **–h** und **–s** für den Befehl **wcollect** angehalten und erneut gestartet werden.

Beispiele

Es folgen Beispiele für verschiedene Operationen, die Sie mit dem Befehl **wcollect** ausführen können.

Aktuelle Konfigurationsdaten anfordern

Im folgenden Beispiel werden aktuelle Konfigurationsdaten für einen Kollektor zurückgegeben:

```
wcollect @ManagedNode:aztlan
```

Ausgabe:

```
Collector: @ManagedNode:aztlan
        debug_level = DEBUG (all messages)
        debug_log_size = 2 MB
        runtime_location = /data/aztlan/aztlan.db/mcollect
        depot_size = 40 MB
        depot_chunk = 1024 KB
        thread_idle_down_time = 60 seconds
        thread_sleep_time = 5 seconds
        max_input_threads = 5
        max_input_retries = 10
        max_output_threads = 5
        retry_delay_time = 1 seconds
        offlinks =
        log_completed_ctoc = true
```

Kollektor nach Abschluss von Datensammlungen stoppen

Mit folgendem Befehl wird ein Kollektor gestoppt, nachdem die Verarbeitung aller aktiven Datensammlungen abgeschlossen ist:

```
wcollect –h graceful @Gateway:drodriguez-gateway
```

Kollektor konfigurieren

Mit folgendem Befehl wird der Umfang der zu protokollierenden Debug-Informationen sowie die Position und Größe des Depots für einen Kollektor angegeben:

```
wcollect –d 3 –l /tmp/dionicio/depot –z 80 @ManagedNode:aztlan
```

Anmerkung: Nachdem dieser Befehl ausgeführt wurde, muss der Kollektor gestoppt werden, und alle Daten müssen aus dem alten Laufzeitverzeichnis in das neue Verzeichnis verschoben werden.

Verbindungen zu einem Kollektor abschalten

Im folgenden Beispiel werden die Verbindungen zwischen dem Kollektor auf 'aztlan' und den Objekt-Dispatchern 2, 5, 6, 7, 8 und 11 abgeschaltet. Anschließend kann der Kollektor auf 'aztlan' von diesen Systemen keine Daten sammeln. (Dieser Befehl hat keinen Einfluss auf die Datensammlung zwischen Endpunkten und dem ersten Gateway-Kollektor.)

```
wcollect –x "2,5-8,11" @ManagedNode:aztlan
```

Alle Verbindungen zu einem Kollektor herstellen

Im folgenden Beispiel werden zu allen Systemen, zu denen die Verbindungen zuvor abgeschaltet wurden, die Verbindungen erneut hergestellt.

```
wcollect –x "" @ManagedNode:aztlan
```

Weitere Informationen

wcstat, **wrpt** (im Handbuch *Tivoli Management Framework Reference Manual*)

wcrtinvcb

Mit diesem Befehl wird das Callback-Objekt von Inventory erstellt.

Syntax

wcrtinvcb *Verwalteter Knoten*

Beschreibung

Mit diesem Befehl wird das Callback-Objekt von Inventory erstellt. Bei der Installation von Tivoli Inventory wird das Callback-Objekt von Inventory automatisch auf dem von Ihnen angegebenen verwalteten Knoten erstellt. Mit diesem Befehl können Sie eine Instanz des Callback-Objekts von Inventory erstellen, beispielsweise um es auf einen anderen verwalteten Knoten zu verschieben. Eine Anleitung dazu finden Sie unter „Das Callback-Objekt von Inventory verschieben” auf Seite 103. Das Callback-Objekt von Inventory kann auf einem beliebigen verwalteten Knoten in Ihrer TMR erstellt werden. Es erfüllt folgende Funktionen:

- Wenn Inventory-Daten nicht mit Hilfe der Kollektorhierarchie gesammelt werden können, sendet MDist 2 Daten an das Callback-Objekt von Inventory, das die Daten an den Inventory Data Handler weiterleitet. Anschließend werden die Daten über eines oder mehrere RIM-Objekte an das Konfigurations-Repository gesendet.
- Bei allen Suchläufen, sendet MDist 2 eine Statusnachricht an das Callback-Objekt von Inventory, mit der angegeben wird, ob das Inventory-Profil erfolgreich an den Endpunkt gesendet wurde. Diese Nachricht bedeutet jedoch nur, dass der Endpunkt das Profil verarbeitet hat, und nicht, dass die Daten des Suchlaufs erfolgreich an das Konfigurations-Repository gesendet wurden.

Optionen

Verwalteter Knoten

Gibt den Namen des verwalteten Knotens an, auf dem Sie das Callback-Objekt von Inventory erstellen möchten.

Autorisierung

admin

Beispiele

Im folgenden Beispiel wird das Callback-Objekt von Inventory auf dem verwalteten Knoten ’fuji’ erstellt.

```
wcrtinvcb fuji
```

wcrtinvdh

Mit diesem Befehl wird eine Instanz des Inventory Data Handler erstellt, und es werden einige seiner Optionen festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wcrtinvdh –d *Statusverzeichnis Verwalteter_Knoten*

wcrtinvdh –n *alle n Minuten bündeln Verwalteter_Knoten*

wcrtinvdh –q *alle n Ziele bündeln Verwalteter_Knoten*

wcrtinvdh –r *Max. Anzahl RDBMS-Einträge Verwalteter_Knoten*

wcrtinvdh –s {**YES** | **NO**} *Verwalteter_Knoten*

wcrtinvdh –t *Verzögerungszeit_RDBMS-Wiederholung Verwalteter-_Knoten*

Beschreibung

Mit diesem Befehl wird eine Instanz des Inventory Data Handler erstellt, das Intervall zum Senden von Nachrichten bei Abschluss der Suchläufe angegeben sowie die Anzahl der Wiederholungen und das Zeitlimit für das Schreiben von Inventory-Daten in das Konfigurations-Repository festgelegt.

Mit dem Befehl **wcrtinvdh** können Sie eine Instanz des Inventory Data Handler mit der Bezeichnung '@InvDataHandler:inv_data _handler' in der Tivoli-Objektdatenbank auf dem von Ihnen angegebenen verwalteten Knoten erstellen. Der Inventory Data Handler ist das Tivoli Inventory-Objekt, das die von den Kollektoren empfangenen Daten an das Konfigurations-Repository sendet. Der Inventory Data Handlern wird bei der Installation von Tivoli Inventory automatisch erstellt. Mit diesem Befehl können Sie eine Instanz des Inventory Data Handler erstellen, beispielsweise um den Inventory Data Handler auf einen anderen verwalteten Knoten zu verschieben. Eine

Anleitung dazu finden Sie unter „Den Inventory Data Handler verschieben” auf Seite 102. In jeder TMR kann immer nur eine Instanz des Inventory Data Handler vorhanden sein.

Außerdem können Sie mit dem Befehl **wcrtinvdh** das Intervall angeben, in dem Nachrichten über den Abschluss von Suchläufen gesendet werden sollen. (Dieses Intervall kann nur für solche Inventory-Profile angegeben werden, in denen der Option **–n** des Befehls **wsetinvglobal** der Wert **BUNDLE** zugeordnet wurde.)

Darüber hinaus können Sie mit dem Befehl **wcrtinvdh** die Anzahl der Wiederholungen und das Zeitlimit für Schreibvorgänge im Konfigurations-Repository festlegen.

Optionen

–d *Statusverzeichnis*

Gibt an, in welchem Verzeichnis der Inventory Data Handler Statusinformationen speichert, die im Falle eines Systemausfalls wiederhergestellt werden können. Standardmäßig befindet es sich im Verzeichnis **$DBDIR/inventory/stat_dir** auf dem verwalteten Knoten, auf dem sich der Inventory Data Handler befindet.

–n *alle n Minuten bündeln*

Gibt das Intervall an, in dem Nachrichten über den Abschluss von Suchläufen gesendet werden sollen (wenn der Option **–n** des Befehls **wsetinvglobal** der Wert **BUNDLE** zugeordnet wurde). Wenn Sie diesen Wert festlegen, sendet Tivoli Inventory eine Nachricht mit einer Liste der Ziele, auf denen in der angegebenen Zeitspanne Suchläufe abgeschlossen wurden. Der Standardwert ist 10 Minuten.

Wenn in der angegebenen Zeitspanne kein Suchlauf abgeschlossen wurde, wird auch keine Nachricht gesendet.

Wenn Sie für diese Option den Wert 0 angeben, erfolgt die Benachrichtigung entsprechend dem Wert, der der Option **–q** des Befehls **wcrtinvdh** oder des Befehls **wsetinvdh** zugeordnet wurde.

–q *alle n Ziele bündeln*

Gibt die maximale Anzahl von Zielen in einem Bündel an. Ein Bündel ist eine Gruppe von Zielen, deren Statusinformationen zu demselben Zeitpunkt gesendet werden. Der Status gibt für jedes Ziel eines bestimmten Inventory-Profils an, ob ein Suchlauf erfolgreich war oder fehlgeschlagen ist.

Der Standardwert ist 10 Ziele.

Mit dieser Option in Verbindung mit der Option **–n** des Befehls **wsetinvglobal** kann angegeben werden, dass die Statusinformationen in Bündeln gesendet werden sollen.

Anmerkungen:

1. Wenn Sie sowohl der Option **–q** *alle n Zielen bündeln* als auch der Option **–n** *alle n Minuten bündeln* den Wert '0' zuweisen, findet keine Bündelung statt, und Sie erhalten erst dann eine Nachricht, wenn die Suchläufe auf allen Zielen abgeschlossen wurden.
2. Wenn Sie sowohl der Option **–q** *alle n Ziele bündeln* als auch der Option **–n** *alle n Minuten bündeln* einen positiven Wert zuordnen, findet jedes Mal eine Bündelung statt, wenn einer der angegebenen Werte (Anzahl der Ziele oder Minuten) erreicht ist.

–r *Max. Anzahl RDBMS-Einträge*

Gibt an, wie oft der Inventory Data Handler versucht, Daten in ein RIM-Objekt zu schreiben. Wenn die maximale Anzahl der Wiederholungen erreicht ist, wird eine Fehlernachricht gesendet.

Der Standardwert ist 5.

–s {YES | NO}

Gibt an, ob der Inventory Data Handler Statusinformationen speichern soll, die im Falle eines Systemausfalls wiederhergestellt werden können. Wenn Sie dieser Option den Wert **YES** zuordnen, werden Statusinformationen gespeichert. Wenn Sie dieser Option den Wert **NO** zuordnen, werden keine Statusinformationen gespeichert. Standardmäßig werden Statusinformationen gespeichert.

–t *Verzögerungszeit RDBMS-Wiederholung*

Gibt einen Wert an, auf dessen Grundlage das Zeitlimit zwischen wiederholten Versuchen, Daten in ein RIM-Objekt zu schreiben, berechnet wird. Das Zeitlimit wird nach dem Algorithmus *Zeitlimit*Wiederholungszähler* berechnet. Beispiel: Bei der ersten Wiederholung mit einem Zeitlimitwert von 30 Sekunden wird der Zeitgeber durch den Algorithmus auf 30 * 1, also 30 Sekunden gesetzt. Bei der zweiten Wiederholung wird der Zeitgeber auf 30 * 2, also 1 Minute gesetzt.

Der Standardwert ist 30 Sekunden.

Verwalteter Knoten

Gibt an, welcher verwaltete Knoten als Inventory Data Handler dienen soll.

Autorisierung

admin, **senior** oder **super**

Beispiele

Im folgenden Beispiel wird der Inventory Data Handler auf dem verwalteten Knoten 'mckinley' erstellt, das Zeitintervall für Nachrichten auf 20 Minuten festgelegt (für Inventory-Profile, in denen der Benachrichtigungsoption der Wert **BUNDLE** zugeordnet wurde; weitere Informationen hierzu finden Sie unter dem Befehl **wsetinvglobal**) und die Anzahl der Wiederholungen für fehlgeschlagene Sendevorgänge an das RIM-Objekt auf 3 festgelegt.

```
wcrtinvdh –n 20 –r 3 mckinley
```

Weitere Informationen

wcollect, **wgetinvdh**, **wgetinvglobal**, **wsetinvdh**, **wsetinvglobal**

wcstat

Mit diesem Befehl werden Statusinformationen zu einem Kollektor, Informationen zu einer CTOC (collection table of contents) sowie der Inhalt der Warteschlangen eines Kollektors zurückgegeben.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wcstat *Kollektor*

wcstat –v *CTOC_ID Kollektor*

wcstat –q [**ioecd**] *Kollektor*

Dabei wird *Kollektor* in einem der folgenden Formate angegeben: **@ManagedNode:***Kollektorname*, **@Gateway:***Kollektorname* oder **@InvDataHandler:inv_data_handler**.

Beschreibung

Mit dem Befehl **wcstat** können Sie folgende Tasks ausführen:

- Statusinformationen zu einem bestimmten Kollektor abrufen.
- Informationen zu einer bestimmten CTOC auf einem Kollektor abrufen.
- Den Inhalt von bestimmten oder allen Warteschlangen eines Kollektors abrufen.

Optionen

Kollektor

Gibt den Namen des Kollektors an, für den Sie den Befehl **wcstat** ausführen möchten. Verwenden Sie dabei eines der folgenden Formate:

- **@Gateway:***Kollektorname*; dabei steht *Kollektorname* für den Namen eines als Kollektor konfigurierten Gateways.

- **@ManagedNode:***Kollektorname*; dabei steht *Kollektorname* für den Namen eines als Kollektor konfigurierten verwalteten Knotens.

–v *CTOC-ID*

Gibt die CTOC (collection table of contents) an, für die Statusinformationen zurückgegeben werden sollen. .

–q *Option*

Gibt die Warteschlangenart an, die zurückgegeben werden soll. Mit folgenden Optionen wird der Inhalt der entsprechenden Warteschlange zurückgegeben. Die Optionen können beliebig miteinander kombiniert werden.

- **i** — Eingabewarteschlange
- **o** — Ausgabewarteschlange
- **e** — Fehlerwarteschlange
- **c** — Warteschlange für beendete Suchläufe
- **d** — Warteschlange für verzögerte Suchläufe

Autorisierung

senior

Beispiele

Es folgen Beispiele für Operationen, die Sie mit dem Befehl **wcstat** ausführen können.

Inhalt einer Warteschlange für beendete Suchläufe zurückgeben

Im folgenden Beispiel wird der Inhalt der abgeschlossenen Warteschlange für den Kollektor 'aztlan' zurückgegeben:

```
wcstat –q c @Gateway:aztlan–gateway
```

Ausgabe:

```
CTOC ID: wepm_ctoc_928780348
CTOC Properties:
    PRIORITY: 1
    COLL_STATUS: TRUE
    SOURCE_NAME: drodriguez2
    SOURCE_OID: 2112331601.2.19
```

```
    SOURCE_METHOD: mc_get_data
    DEST_OID: 2112331601.1.675
    INV_DDC::InvDataHandler
    DEST_METHOD: mc_request_collection
    DATAPACK: 2129
Client Properties:
    scan_id: 2147483647
Collection Status: CTOC_DONE
#Retries: 0
```

Statusinformationen einer CTOC zurückgeben

Im folgenden Beispiel werden Informationen zur angegebenen CTOC zurückgegeben:

```
wcstat –v ctoc3_11836_9110 @ManagedNode:calypso
```

Ausgabe:

```
CTOC ID: ctoc3_11836_9110
CTOC Properties:
    PRIORITY: 1
    COLL_STATUS: OK
    SOURCE_OID: 1637823410.2.19
    DEST_OID: 1637823410.1.552#MCFTP::Server#
    DEST_METHOD: callback_method
    DATAPACK: 33637364
Client Properties:
    MC_DEST_DIR: /tmp
Collection Status: QUEUED_OUTPUT
#Retries: 0
```

Weitere Informationen

wcollect

wdistinv

Mit diesem Befehl werden Inventory-Profile verteilt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wdistinv –l *Mdist2-Token=Wert* **@InventoryConfig:***Profilname*

wdistinv –T *Subskribentendatei* **@InventoryConfig:***Profilname*

wdistinv @InventoryConfig:*Profilname* **@Endpoint:***Ziel1 ...*

Beschreibung

Mit dem Befehl **wdistinv** können Sie über die Befehlszeile ein Inventory-Profil verteilen. Wenn Sie keine Ziele für die Verteilung angeben, wird das Inventory-Profil an alle Subskribenten des Profilmanagers verteilt, in dem das Inventory-Profil gespeichert ist.

Optionen

–l *Mdist2_Token=Wert*

Gibt einen Wert für die MDist 2-Verteilungsoptionen an. Sie können mehrere Optionen angeben, indem Sie den Ausdruck **–l** *Mdist2_Token=Wert* mehrfach angeben. Sie können folgende Token verwenden:

label=*Beschreibung*

Gibt eine Zeichenfolge als Beschreibung der Verteilung an. Der Standardwert ist der Name des Inventory-Profils.

priority=h | m | l

Gibt die Prioritätsstufe an, d. h. die Reihenfolge, in der Verteilungen von Repeatern vorgenommen werden. Sie können einen der folgenden Werte für die Prioritätsstufe angeben:

h Hohe Priorität.

m Mittlere Priorität. Dies ist der Standardwert.

l Niedrige Priorität.

notify_interval=*Minuten*
Gibt das Intervall in Minuten an, in dem jeder Repeater die Ergebnisse der MDist 2-Verteilung bündelt und zurückgibt. Als Wert muss eine positive ganze Zahl angegeben werden. Der Standardwert ist 1 Minute.

send_timeout=*Sekunden*
Gibt die Zeitspanne an, die der Repeater dem Zielsystem für den Empfang eines Datenblocks einräumt (z. B. ein Repeater, der ein Inventory-Profil an einen Endpunkt verteilt). Das Zeitlimit dient zum Auffinden von Netzwerk- oder Endpunktfehlern. Wenn ein Zeitlimit überschritten wird, bleibt die Verteilung in der Warteschlange des Repeaters bestehen, und nach Ablauf des mit dem in Tivoli Management Framework verfügbaren Befehls **wmdist** festgelegten Wertes **conn_retry_interval** wird ein neuer Versuch gestartet.

execute_timeout=*Sekunden*
Gibt die Zeitspanne an, die ein Kollektor darauf wartet, dass ein Endpunkt die Ergebnisse zurückgibt (die Rückgaben auf abwärts gerichtete Aufrufe), nachdem alle Daten gesendet wurden.

deadline=*MM/TT/JJJJ hh:mm*
Gibt das Datum an, an dem eine Verteilung abläuft (danach schlägt sie für bis dahin nicht verfügbare Ziele fehl). Das Standarddatum ist 72 Stunden nach dem Zeitpunkt, zu dem das Profil verteilt wurde.

mandatory_date=*MM/TT/JJJJ hh:mm*
Gibt das Datum an, bis zu dem die Verteilung ausgeführt sein muss. Verteilungen an Ziele für standortunabhängigen Zugriff (z. B. Laptops) können bis zu diesem Datum verzögert werden. Wenn das Datum erreicht ist, wird das Profil automatisch an alle Ziele für standortunabhängigen Zugriff verteilt, die das

Profil bis dahin noch nicht akzeptiert haben. Der Standardwert ist 60 Stunden nach dem Zeitpunkt, zu dem das Profil verteilt wurde.

force_mandatory=y | n

Gibt an, wie mit verbindlichen Verteilungen auf Endpunkten für standortunabhängigen Zugriff umgegangen wird, nachdem das verbindliche Datum überschritten wurde. Sie können eine der folgenden Optionen auswählen:

- **y** - Die verbindliche Verteilung wird automatisch gestartet, sobald der Benutzer mit standortunabhängigem Zugriff eine Verbindung herstellt. Dies ist die Standardoption.
- **n** - Der Benutzer mit standortunabhängigem Zugriff kann sich dafür entscheiden, die verbindliche Verteilung nicht zu starten. Er kann jedoch keine anderen Operationen ausführen, bis er die verbindliche Verteilung ausgeführt hat.

escalate_date_*n***=***MM/TT/JJJJ hh:mm*

Gibt das Datum an, an dem eine Erinnerungsnachricht an die Ziele mit standortunabhängigem Zugriff gesendet werden soll, die das Profil noch nicht empfangen haben. Für den Wert *n* können Sie eine Zahl von 0 bis 9 angeben, so dass Sie bis zu zehn verschiedene Nachrichten erstellen können. Die Zahlen für den Wert *n* müssen in aufsteigender numerischer Reihenfolge angegeben werden und dabei darf keine Zahl übersprungen werden. Für jedes Sendedatum muss außerdem mit Hilfe des Tokens **escalate_msg** eine Dringlichkeitsnachricht eingegeben werden.

escalate_msg_*n*

Gibt einen Nachrichtentext an, der an dem zugeordneten Sendedatum an die Ziele mit standortunabhängigem Zugriff gesendet werden soll, die das Profil noch nicht erhalten haben. Für den Wert *n* können

Sie eine Zahl von 0 bis 9 angeben, so dass Sie bis zu zehn verschiedene Nachrichten erstellen können. Die Zahlen für den Wert *n* müssen in aufsteigender numerischer Reihenfolge angegeben werden und dabei darf keine Zahl übersprungen werden. Für jede Dringlichkeitsnachricht muss außerdem mit Hilfe des Tokens **escalate_date** ein Sendedatum angegeben werden. Geben Sie Nachrichten in folgendem Format ein:

```
escalate_msg_n="Nachrichtentext"
```

hidden=y | n

Gibt an, ob eine Verteilung an einen Endpunkt mit standortunabhängigem Zugriff ausgeblendet werden soll. Ausgeblendete Verteilungen werden automatisch an den Endpunkt für standortunabhängigen Zugriff verteilt und werden für den Benutzer dieses Endpunkts nicht sichtbar ausgeführt. Verteilungen, die nicht ausgeblendet wurden, sind für den Benutzer mit standortunabhängigem Zugriff sichtbar, und ihre Ausführung kann von ihm verzögert werden. Sie können eine der folgenden Optionen verwenden:

- **y** - Die Verteilung wird ausgeblendet.
- **n** - Die Verteilung wird nicht ausgeblendet.

Wenn für dieses Argument der Wert 'y' angegeben wird, müssen für das verbindliche Datum, das Sendedatum und den Nachrichtentext keine Werte angegeben werden.

roam_endpoints=y | n

Gibt an, ob die im Befehl **wdistinv** definierte Operation mobile Endpunkte bzw. Endpunkte für standortunabhängigen Zugriff (wie z. B. Laptops) unterstützt. Sie können eine der folgenden Optionen verwenden:

- **y** - Die Verteilung soll an ein beliebiges Gateway übertragen werden, zu dem der Endpunkt für standortunabhängigen Zugriff eine Verbindung herstellt.
- **n** - Nachdem ein Profil auf einem Gateway in eine Warteschlange eingereiht wurde, kann es zu keinem anderen Gateway übertragen werden.

wake_on_lan=y | n

Gibt an, ob die Operation eine WoL-Nachricht (Wake-on-LAN) senden soll, damit ein System, das zum Zeitpunkt der Verteilung eines Inventory-Profils nicht verfügbar ist, gestartet wird. Diese Option funktioniert nur auf Systemen, deren Netzwerkkarte WoL-fähig ist. Sie können eine der folgenden Optionen verwenden:

- **y** — Es wird eine WoL-Nachricht gesendet.
- **n** — Es wird keine WoL-Nachricht gesendet.

–T *Subskribentendatei*

Gibt eine Textdatei an, die eine Liste der Verteilungsziele enthält. In dieser Datei können Endpunkte, Profilmanager oder beides aufgelistet sein. Die Ziele müssen in der Datei in folgendem Format aufgeführt werden:

- **@Endpoint:***Endpunktname*
- **@ProfileManager:***Profilmanagername*

Für den Wert *Subskribentendatei* können die Dateien sowohl mit einem relativen als auch mit einem absoluten Pfadnamen angegeben werden. Sie können einen oder mehrere Dateinamen angeben. Mehrere Dateinamen werden jeweils durch ein Leerzeichen voneinander getrennt. Sie können diese Option entweder allein verwenden oder mit Hilfe der Option **@Endpoint:***Ziel* zusätzlich eines oder mehrere Ziele für die Verteilung angeben.

@InventoryConfig:*Profilname*

Gibt das Profil an, für das Sie den Befehl **wdistinv** ausführen möchten.

@Endpoint:*Ziel*

Gibt den Endpunkt an, an den Sie das Inventory-Profil verteilen möchten.

Autorisierung

Inventory_scan, **senior** oder **super**

Beispiele

Im folgenden Beispiel wird für die Verteilung des Profils 'InvProfile' eine Beschreibung angegeben und ein verbindliches Datum festgelegt:

```
wdistinv -l label="Inventory-Verteilung" \
-l mandatory_date=04/06/2001 12:00 @InventoryConfig:InvProfile
```

Im folgenden Beispiel wird eine Datei angegeben, die eine Liste der Ziele enthält, und die Priorität für die Verteilung des Profils 'InvProfile' festgelegt:

```
wdistinv -T /tmp/endpoint.list -l priority=high \
@InventoryConfig:InvProfile
```

Weitere Informationen

wdistrib, **wmdist**, **wrpt** (im Handbuch *Tivoli Management Framework Reference Manual*)

wepscan

Mit diesem Befehl werden Suchläufe auf einem Endpunkt eingeleitet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wepscan

wepscan –c

wepscan –d

wepscan –s

Beschreibung

Mit dem Befehl **wepscan** können Sie einen Suchlauf von einem Endpunkt aus einleiten, anstatt von einem fernen System aus ein Inventory-Profil an den Endpunkt verteilen zu müssen. Sie können angeben, ob der Suchlauf auf dem Endpunkt ausgeführt werden soll, ob die Suchlaufdaten an das Konfigurations-Repository gesendet werden sollen oder ob beide Vorgänge ausgeführt werden sollen. Mit dem Befehl **wepscan** kann auch eine Protokolldatei erstellt werden, die die Suchlaufdaten oder den Inhalt der Konfigurationsdatei enthält.

Bevor Sie den Befehl **wepscan** verwenden können, müssen Sie mit Hilfe der Einstellungen für Ihre Umgebung den für diesen Befehl erforderlichen Zugriff auf gemeinsam benutzte Bibliotheken ermöglichen. Führen Sie dazu das Skript 'lcf_env' aus. Je nachdem, auf welcher Plattform der Endpunkt ausgeführt wird, trägt das Skript eine der folgenden Bezeichnungen: lcf_env.bat, lcf_env.cmd, lcf_env-.csh, oder lcf_env.sh. Das Skript befindet sich entweder direkt in dem Verzeichnis, in dem der Endpunkt installiert ist, oder im Unterverzeichnis 'dat/1' dieses Verzeichnisses.

Solaris-Benutzer müssen vor der Verwendung des Befehls **wepscan** der Umgebungsvariablen 'LD_LIBRARY_PATH' einen Wert zuordnen. Benutzer der Bourne- oder Korn-Shell müssen beispielsweise den Wert der Umgebungsvariable wie folgt festlegen:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:.
export LD_LIBRARY_PATH
```

HP-UX-Benutzer müssen vor der Verwendung des Befehls **wepscan** der Umgebungsvariablen 'SHLIB_PATH' einen Wert zuordnen. Benutzer der Bourne- oder Korn-Shell müssen beispielsweise den Wert der Umgebungsvariable wie folgt festlegen:

```
SHLIB_PATH=$SHLIB_PATH:.
export SHLIB_PATH
```

Der Befehl **wepscan** muss im Verzeichnis '$LCF_BASE_DIR/inv/SCAN' ausgeführt werden, wobei '$LCF-_BASE_DIR' für das Verzeichnis steht, in dem Sie den Endpunkt installiert haben.

Wenn Sie den Befehl **wepscan** ausführen, ohne Optionen anzugeben, wird die letzte Version der Konfigurationsdatei 'config.dmp', die an den Endpunkt verteilt wurde, gelesen und anschließend der Suchlauf gemäß den in der Konfigurationsdatei angegebenen Optionen ausgeführt. Danach wird eine MIF-Datei (Management Information Format) erstellt, diese MIF-Datei syntaktisch analysiert und eine codierte und komprimierte Datendatei mit dem Namen 'INV_SA-.DAT' erstellt. Die Datendatei wird auf dem Endpunkt gespeichert.

Optionen

–c Liest den Inhalt der Konfigurationsdatei 'config.dmp' und schreibt ihn in eine Protokolldatei mit dem Namen 'sa_config.log'. Diese Protokolldatei wird in dem Verzeichnis gespeichert, in dem Sie den Befehl **wepscan** ausgeführt haben.

–d Führt einen Suchlauf aus, speichert die Suchlaufdaten auf dem Endpunkt und erstellt eine Protokolldatei mit dem Namen 'sa_results.log', die die Suchlaufdaten enthält.

Diese Protokolldatei wird in dem Verzeichnis gespeichert, in dem Sie den Befehl **wepscan** ausgeführt haben. Bei Verwendung der Option **–d** wird nicht nur die Datei 'sa_results.log', sondern zusätzlich auch die Datei 'sa_config.log' erstellt, die auch einzeln durch Verwendung der Option **–c** erstellt werden kann.

–s Sendet die Suchlaufdaten (d. h. die Datei 'INV_SA-.DAT') an das Konfiguratins-Repository.

Autorisierung

Dieser Befehl wird lokal auf dem Endpunkt ausgeführt. Daher ist keine Tivoli-Berechtigung erforderlich.

Hinweise

Dieser Befehl kann auf Endpunkten, die auf NetWare- oder OS/2-Gateways ausgeführt werden, nicht verwendet werden.

Beispiele

Im folgenden Beispiel wird auf einem Endpunkt ein Suchlauf anhand der in der Konfigurationsdatei des Endpunkts angegebenen Suchlaufoptionen ausgeführt, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert:

```
wepscan
```

Im folgenden Beispiel werden die auf dem Endpunkt gespeicherten Suchlaufdaten an das Konfigurations-Repository gesendet:

```
wepscan -s
```

Im folgenden Beispiel wird der Inhalt der Konfigurationsdatei 'config.dmp' in die Datei 'sa_config.log' geschrieben:

```
wepscan -c
```

wgetinvdh

Mit diesem Befehl werden Konfigurationsdaten des Inventory Data Handler zurückgegeben.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvdh –a

wgetinvdh –d

wgetinvdh –n

wgetinvdh –q

wgetinvdh –r

wgetinvdh –s

wgetinvdh –t

Beschreibung

Mit dem Befehl **wgetinvdh** werden Informationen zum Inventory Data Handler zurückgegeben, darunter auch Angaben zur Benachrichtigung, zur Anzahl der Wiederholungen und zum Zeitlimit für das Schreiben von Inventory-Daten in ein RIM-Objekt.

Optionen

–a Gibt alle verfügbaren Informationen zum Inventory Data Handler zurück.

–d Gibt den Wert der Option *Statusverzeichnis*, der im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird, zurück. In diesem Verzeichnis werden Statusinformationen gespeichert, damit sie im Falle eines Systemausfalls wiederhergestellt werden können.

–n Gibt den Wert der Option *alle n Minuten bündeln,* der im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird, zurück. Der Wert gibt das Intervall in Minuten an, in dem Hinweise über den Abschluss von Suchläufen gesendet werden.

–q Gibt den Wert der Option *alle n Ziele bündeln* zurück, der im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird. Der Wert gibt die maximale Anzahl von Zielen in einem Bündel an.

Ein Bündel ist eine Gruppe von Zielen, deren Statusinformationen zu demselben Zeitpunkt an die Tivoli Inventory-Hinweisgruppe gesendet werden. Der Status gibt für jedes Ziel eines bestimmten Inventory-Profils an, ob ein Suchlauf erfolgreich war oder fehlgeschlagen ist. Mit dieser Option, die im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird, in Verbindung mit der Option **–n**, die im Befehl **wsetinvglobal** angegeben wird, wird festgelegt, wie Statusinformationen in Bündeln gesendet werden sollen.

–r Gibt den Wert der Option *Max. Anzahl RDBMS-Einträge*, der im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird, zurück. Der Wert gibt die Anzahl der Wiederholungen für den Versuch des Inventory Data Handler an, Daten in ein RIM-Objekt zu schreiben, bevor für das Ziel, dem die Daten zugeordnet sind, eine Fehlernachricht zurückgegeben wird.

–s Gibt an, ob der Inventory Data Handler für den Fall eines Systemausfalls Statusinformationen speichert. Wenn der Wert 'YES' zurückgegeben wird, bedeutet dies, dass Statusinformationen gespeichert werden. Wenn der Wert 'NO' zurückgegeben wird, bedeutet dies, dass keine Speicherung von Statusinformationen stattfindet.

–t Gibt den Wert der Option *Verzögerungszeit RDBMS-Wiederholung*, der im Befehl **wcrtinvdh** oder **wsetinvdh** angegeben wird, zurück. Der Wert wird zur Berechnung des Zeitlimits in Sekunden zwischen Schreibvorgängen in ein RIM-Objekt verwendet. Das Zeitlimit wird nach dem Algorithmus *Zeitlimit*Wiederholungszähler* berechnet. Beispiel: Bei der ersten Wiederholung mit einem Zeitlimitwert von 30 Sekunden wird der Zeitgeber durch den Algorithmus auf 30 * 1, also 30 Sekunden gesetzt. Bei der zweiten Wiederholung wird der Zeitgeber auf 30 * 2, also 1 Minute gesetzt.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie den Befehl ausführen, ohne Optionen anzugeben, werden alle Konfigurationsdaten des Inventory Data Handler zurückgegeben.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen zum Inventory Data Handler zurückgegeben:

```
wgetinvdh –a
```

Ausgabe:

```
Save status in the directory: /Tivoli/db/fuji.db/inventory/stat_dir
Send bundled notification every: 10 minutes
Send bundled notification every: 10 targets
Max retries for RDBMS errors:      5
Save status in case of failure:  YES
Retry delay time for RDBMS errors: 30 seconds
Managed Node:                      fuji
```

Weitere Informationen

wcollect, **wcrtinvdh**, **wsetinvdh**, **wsetinvglobal**

wgetinvglobal

Mit diesem Befehl werden die für ein Inventory-Profil festgelegten allgemeinen Merkmale aufgelistet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvglobal *Profilname*

wgetinvglobal –d *Profilname*

wgetinvglobal –e *Profilname*

wgetinvglobal –f *Profilname*

wgetinvglobal–h *Profilname*

wgetinvglobal –l *Profilname*

wgetinvglobal –m *Profilname*

wgetinvglobal –n *Profilname*

wgetinvglobal –t *Profilname*

wgetinvglobal –u *Profilname*

wgetinvglobal –w *Profilname*

Beschreibung

Mit dem Befehl **wgetinvglobal** können Sie die für ein Inventory-Profil festgelegten allgemeinen Merkmale anzeigen. Welche Informationen mit diesem Befehl zurückgegeben werden, hängt davon ab, welche Optionen Sie bei dessen Ausführung angeben.

Optionen

–d Gibt Informationen zu den Verteilungsoptionen des

angegebenen Inventory-Profils zurück. Folgende Werte können zurückgegeben werden:

CONFIG
: Nur die Konfigurationsdatei verteilen.

Alle
: Konfigurationsdatei verteilen und den Suchlauf ausführen.

–e
: Gibt die Zeit in Sekunden zurück, die dem Profil für den Suchlauf auf einem Endpunkt eingeräumt werden, bevor es abläuft.

–f
: Gibt den Namen der Protokolldatei zurück, in der die Statusinformationen für Tivoli Inventory gespeichert werden.

–h
: Gibt den Namen des verwalteten Knotens zurück, auf dem die Statusinformationen für Tivoli Inventory gespeichert werden.

–l
: Gibt einen der folgenden Werte zurück, mit denen angegeben wird, wo die Statusinformationen für Tivoli Inventory protokolliert werden:

  NOTICE_GROUP
  : Statusinformationen werden an die Tivoli Inventory-Hinweisgruppe gesendet.

  LOG_FILE
  : Statusinformationen werden in der Protokolldatei gespeichert, die mit Hilfe der Optionen **–h** und **–l** des Befehls **wsetinvglobal** angegeben wurde.

  TEC
  : Statusinformationen werden an die TEC-Konsole (TEC - Tivoli Enterprise Console) gesendet.

  OFF
  : Es werden keine Statusinformationen gespeichert.

–m
: Gibt einen Wert zurück, mit dem angegeben wird, ob eine Verteilung an einen mobilen Endpunkt bzw. Endpunkt für standortunabhängigen Zugriff (wie z. h. einen Laptop) ausgeblendet sein soll. Ausgeblendete Verteilungen werden automatisch an den

Endpunkt für standortunabhängigen Zugriff verteilt und werden für den Benutzer dieses Endpunkts nicht sichtbar ausgeführt. Verteilungen, die nicht ausgeblendet wurden, sind für den Benutzer mit standortunabhängigem Zugriff sichtbar, und ihre Ausführung kann von ihm verzögert werden. Einer der folgenden Werte kann zurückgegeben werden:

Y Die Verteilung wird ausgeblendet.

N Die Verteilung wird nicht ausgeblendet.

–n Gibt Informationen darüber zurück, wann ein Hinweis auf den Abschluss eines Suchlaufs auf den jeweiligen Zielen gesendet wird. Folgende Werte können zurückgegeben werden:

IMMEDIATE
Gibt an, dass der Hinweis unmittelbar nach Abschluss des Suchlaufs auf jedem einzelnen Ziel gesendet wird.

DONE
Gibt an, dass der Hinweis erst gesendet wird, nachdem die Suchläufe auf allen Zielen abgeschlossen wurden.

BUNDLE
Gibt an, dass der Hinweis in regelmäßigen Abständen gemäß den Einstellungen für den Inventory Data Handler gesendet wird. Weitere Informationen zum Bündeln finden Sie unter den Befehlen **wcrtinvdh**, **wgetinvdh** und **wsetinvdh**.

–t Gibt Informationen darüber zurück, welche Hinweisart für ein bestimmtes Ziel gesendet wird, je nachdem, mit welchem Status ein Suchlauf abgeschlossen wurde. Folgende Werte können zurückgegeben werden:

NONE
Gibt an, dass kein Hinweis gesendet wird.

SUCCESS
Gibt an, dass der Hinweis für Ziele gesendet wird, auf denen der Suchlauf erfolgreich abgeschlossen wurde.

FAIL Gibt an, dass der Hinweis für Ziele gesendet wird, auf denen der Suchlauf fehlgeschlagen ist.

All Gibt an, dass der Hinweis für alle Ziele gesendet wird.

–u Gibt Informationen zu den Datenoptionen dieses Profils zurück. Folgende Werte können zurückgegeben werden:

DIFFS
Aktualisierung anhand von Änderungen. Nur seit dem letzten Suchlauf hinzugefügte oder geänderte Daten werden im Konfigurations-Repository gespeichert.

REPLACE
Durch aktuelle Ergebnisse ersetzen. Die Daten im Konfigurations-Repository werden durch die beim Suchlauf gesammelten Daten ersetzt.

–w Gibt an, ob Systeme mit einer WoL-fähigen Netzwerkkarte (Wake-on-LAN) bei der Profilverteilung gestartet werden. Folgende Werte können zurückgegeben werden:

Y Das System wird bei der Profilverteilung gestartet.

N Das System wird bei der Profilverteilung nicht gestartet.

Profilname Gibt das Inventory-Profil an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie den Befehl ausführen, ohne Optionen anzugeben, werden alle verfügbaren Informationen zu den allgemeinen Merkmalen des Inventory-Profils zurückgegeben.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen zu den allgemeinen Merkmalen des Inventory-Profils 'TestProfil' zurückgegeben:

```
wgetinvglobal @InventoryConfig:TestProfil
```

Ausgabe:

```
Distribution option: Distribute configuration file and run scan (ALL)

Log file pathname:   /usr/logs/TestProfile.log
Log file host:       fuji
Notice location:     NOTICE_GROUP
Notice interval:     DONE
Notice type:         ALL
Data option:         Replace with current results (REPLACE)
```

Im folgenden Beispiel wird die Adresse zurückgegeben, an die der Hinweis gesendet werden soll, sowie das Intervall, in dem dieser Vorgang stattfindet:

```
wgetinvglobal -l -n @InventoryConfig:MyProf
```

Ausgabe:

```
Notice location:     NOTICE_GROUP
Notice interval:     DONE
```

Weitere Informationen

wgetinvpcfiles, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixhw**, **wgetinvunixsw**, **wsetinvglobal**

wgetinvpcfiles

Mit diesem Befehl werden Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, aufgelistet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvpcfiles *Profilname*

wgetinvpcfiles –d *Profilname*

wgetinvpcfiles –f *Profilname*

wgetinvpcfiles –m *Profilname*

wgetinvpcfiles –s *Profilname*

Beschreibung

Mit dem Befehl **wgetinvpcfiles** werden Informationen über ein angegebenes Profil bezüglich zu durchsuchender Dateien und Verzeichnisse, auszuführender Skripts und benutzerdefinierter MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, aufgelistet. Folgende Informationen zu einem Profil und dessen Optionen können abgerufen werden:

- Die Verzeichnisse, die bei einem Softwaresuchlauf eingeschlossen oder ausgeschlossen werden.
- Die Dateien oder Dateitypen, die bei einem Softwaresuchlauf eingeschlossen oder ausgeschlossen werden.
- Die MIF-Dateien, die bei einem Suchlauf gelesen werden sollen.
- Die Skripts, die bei einem Suchlauf ausgeführt werden sollen.

Optionen

–d	Gibt eine Liste der Verzeichnisse zurück, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
–f	Gibt eine Liste der Dateien oder Dateitypen zurück, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
–m	Gibt eine Liste der MIF-Dateien zurück, die beim Suchlauf gelesen werden sollen.
–s	Gibt eine Liste der Skripts zurück, die beim Suchlauf auf den Zielen ausgeführt werden sollen.
Profilname	Gibt den Namen des Inventory-Profils an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie den Befehl **wgetinvpcfiles** ausführen, ohne Optionen anzugeben, werden alle verfügbaren Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die durch das angegebene Inventory-Profil gesammelt werden sollen, aufgelistet.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die durch das Inventory-Profil 'MeinProfil' gesammelt werden sollen, aufgelistet:

```
wgetinvpcfiles @InventoryConfig:MeinProfil
```

Die Ausgabe dieses Befehls ist wie folgt:

```
No Include Directories
Exclude Directories:
        */TMP/
        */TEMP/
Include File Types:
        *.EXE
        *.DLL
        *.COM
        *.NLM
No Custom Mif Files
No script to run after scan
```

Weitere Informationen

wgetinvglobal, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixhw**, **wgetinvunixsw**, **wsetinvpcfiles**

wgetinvpchw

Mit diesem Befehl werden die für den Hardwaresuchlauf auf einem PC-Endpunkt festgelegten Optionen aufgelistet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvpchw *Profilname*

wgetinvpchw –d *Profilname*

wgetinvpchw –t *Profilname*

wgetinvpchw–u *Profilname*

Beschreibung

Mit dem Befehl **wgetinvpchw** werden die Optionen für PC-Hardwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden. Folgende Informationen zu einem Profil und dessen Optionen können abgerufen werden:

- Ob der Hardwaresuchlauf von dem Profil ausgeführt wird, ob die Suchlaufdaten an das Konfigurations-Repository gesendet werden oder ob beide Vorgänge ausgeführt werden.
- Nach welchen Hardwarekomponenten beim Suchlauf gesucht wird.

Optionen

–d Listet alle Hardwarekomponenten auf, nach denen gesucht werden kann, und gibt an, ob das Inventory-Profil für die Suche nach den jeweiligen Komponenten konfiguriert wurde.

–t Gibt einen Wert zurück, der anzeigt, ob die Tivoli Hardware-Suchfunktion aktiviert wurde. Folgende Werte können zurückgegeben werden:

YES Die Tivoli Hardware-Suchfunktion wurde aktiviert.

NO Die Tivoli Hardware-Suchfunktion wurde nicht aktiviert.

–u Gibt einen Wert zurück, der anzeigt, ob die beim Hardwaresuchlauf gesammelten Daten der PC-Systeme an das Konfigurations-Repository gesendet werden sollen. Folgende Werte können zurückgegeben werden:

YES Die Suchlaufdaten werden nach Abschluss des Suchlaufs an das Konfigurations-Repository gesendet.

NO Es werden keine Suchlaufdaten an das Konfigurations-Repository gesendet. Sie werden stattdessen auf dem Endpunkt gespeichert.

Profilname
Gibt das Inventory-Profil an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über die Optionen für PC-Hardwaresuchläufe zurückgegeben, die für das Inventory-Profil 'MeinProfil' konfiguriert wurden:

```
wgetinvpchw @InventoryConfig:MeinProfil
```

Ausgabe:

```
Run Tivoli Scanner: YES
Update the hardware scan results in the configuration repository: YES
Data to scan:
        Processor: YES
        Memory: YES
        Operating System: YES
        Storage: YES
        IP Address: YES
        Network Adapter: YES
        Partition: YES
        PC System Params: YES
        Pointing Device: YES
        Keyboard: YES
```

```
IPX Address: YES
Video: YES
Printer: YES
USB Device: YES
```

Im folgenden Beispiel wird angegeben, ob das Inventory-Profil 'MeinProfil' für die Ausführung der Tivoli Hardware-Suchfunktion konfiguriert wurde:

```
wgetinvpchw -t @InventoryConfig:MeinProfil
```

Ausgabe:

```
Run Tivoli Scanner: YES
```

Weitere Informationen

wgetinvglobal, **wgetinvpcfiles**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixhw**, **wgetinvunixsw**, **wsetinvpchw**

wgetinvpcsw

Mit diesem Befehl werden die Optionen für PC-Softwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvpcsw *Profilname*

wgetinvpcsw –b *Profilname*

wgetinvpcsw –c *Profilname*

wgetinvpcsw –f *Profilname*

wgetinvpcsw –h *Profilname*

wgetinvpcsw –r *Profilname*

wgetinvpcsw –s *Profilname*

wgetinvpcsw –x *Profilname*

Beschreibung

Mit dem Befehl **wgetinvpcsw** werden die Optionen für PC-Softwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden. Dieser Befehl gibt Informationen darüber zurück, ob ein Profil für die Ausführung folgender Aktionen konfiguriert wurde:

- Nach grundlegenden Dateiinformationen suchen.
- Für die durchsuchten Dateien Kontrollsummen generieren.
- Angepassten Filter für Suchläufe nach grundlegenden Informationen verwenden.
- Nach Header-Informationen suchen.
- Windows-Registrierungsdatenbank nach Informationen zu installierten Produkten durchsuchen.

- Durchsuchte Dateien mit der Liste der Softwaresignaturen vergleichen.
- Nur ausführbare Dateien durchsuchen.

Optionen

–b Gibt Informationen über Suchläufe nach grundlegenden Dateiinformationen zurück. Folgende Werte können zurückgegeben werden:

SCAN Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Der Endpunkt wird nicht durchsucht. Stattdessen werden die Daten eines früheren Suchlaufs nach grundlegenden Dateiinformationen gesammelt und an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach grundlegenden Dateiinformationen ausgeführt. Es werden keine Daten für grundlegende Dateiinformationen gesammelt.

–c Gibt Informationen zur Konfiguration der Kontrollsummenoptionen zurück. Folgende Werte können zurückgegeben werden:

NONE
Beim Suchlauf werden keine Kontrollsummenwerte der durchsuchten Dateien generiert.

QUICK
Beim Suchlauf wird die schnelle Kontrollsumme der durchsuchten Dateien gesammelt.

FULL Beim Suchlauf wird die vollständige Kontrollsumme der durchsuchten Dateien gesammelt.

MD5 Beim Suchlauf wird die MD5-Kontrollsumme der durchsuchten Dateien gesammelt.

–f Gibt Informationen darüber zurück, ob der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen verwendet wird. Folgende Werte können zurückgegeben werden:

YES Beim Suchlauf wird der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen verwendet.

NO Beim Suchlauf wird der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen nicht verwendet.

–h Gibt Informationen über Suchläufe nach Header-Informationen zurück. Folgende Werte können zurückgegeben werden:

SCAN Der Endpunkt wird nach Header-Informationen durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Der Endpunkt wird nicht durchsucht. Stattdessen werden die Daten eines früheren Suchlaufs nach Header-Informationen gesammelt und an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach Header-Informationen durchsucht, und die Suchlaufdaten

werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach Header-Informationen ausgeführt. Es werden keine Daten für Header-Informationen gesammelt.

–r Gibt Informationen über Suchläufe in der Windows-Registrierungsdatenbank nach Informationen zu installierten Produkten zurück. Folgende Werte können zurückgegeben werden:

SCAN Die Windows-Registrierdatenbank wird nach Informationen zu installierten Produkten durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Der Endpunkt wird nicht durchsucht. Stattdessen werden die Daten eines früheren Suchlaufs nach Informationen in der Registrierungsdatenbank gesammelt und an das Konfigurations-Repository gesendet.

BOTH
Die Windows-Registrierdatenbank wird nach Informationen zu installierten Produkten durchsucht, und die Suchlaufdaten werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach Informationen in der Registrierdatenbank ausgeführt. Es werden keine Daten über Informationen in der Registrierdatenbank gesammelt.

–s Vergleicht durchsuchte Dateien mit der Liste der Softwaresignaturen. Folgende Werte können zurückgegeben werden:

SCAN Durchsuchte Dateien werden mit der Liste

der Softwaresignaturen verglichen. Daten für übereinstimmende Dateien werden auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf nach übereinstimmenden Softwaresignaturen ausgeführt. Stattdessen werden die Daten eines früheren Suchlaufs nach übereinstimmenden Softwaresignaturen gesammelt und an das Konfigurations-Repository gesendet.

BOTH
Durchsuchte Dateien werden mit der Liste der Softwaresignaturen verglichen, und die Daten für übereinstimmende Dateien werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach übereinstimmenden Softwaresignaturen ausgeführt. Es werden keine Daten für übereinstimmende Softwaresignaturen gesammelt.

–x Durchsucht nur ausführbare Dateien. Folgende Werte können zurückgegeben werden:

YES Es werden nur ausführbare Dateien durchsucht.

NO Der Suchlauf ist nicht auf ausführbare Dateien beschränkt.

Profilname Gibt den Namen des Inventory-Profils an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@Inventory-Config:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie diesen Befehl ausführen, ohne Optionen anzugeben, werden alle verfügbaren Informationen über die Optionen für PC-Softwaresuchläufe zurückgegeben, die für ein bestimmtes Inventory-Profil konfiguriert wurden.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über die Optionen für PC-Softwaresuchläufe zurückgegeben, die für das Inventory-Profil 'MeinProfil' konfiguriert wurden:

```
wgetinvpcsw @InventoryConfig:MeinProfil
```

Von diesem Befehl wird Folgendes zurückgegeben:

```
Scan for and update basic file information: NO
Cyclic Redundancy Check: No CRC
Scan for and update header information: BOTH
Scan for and update PC registry information: BOTH
Scan for and update matching software signatures: NO
Apply custom filter to basic file information scan results: NO
Executable files only: YES
```

Im folgenden Beispiel wird die Kontrollsummenoption, die für das Inventory-Profil 'MeinProfil' konfiguriert wurde, zurückgegeben:

```
wgetinvpcsw -c @InventoryConfig:MeinProfil
```

Von diesem Befehl wird Folgendes zurückgegeben:

```
Cyclic Redundancy Check: No CRC
```

Weitere Informationen

wgetinvglobal, **wgetinvpcfiles**, **wgetinvpchw**, **wgetinvunixfiles**, **wgetinvunixhw**, **wgetinvunixsw**, **wsetinvpcsw**

wgetinvunixfiles

Mit diesem Befehl werden Informationen über das angegebene Inventory-Profil bezüglich zu durchsuchender Dateien und Verzeichnisse, auszuführender Skripts

und benutzerdefinierter MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, aufgelistet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvunixfiles *Profilname*

wgetinvunixfiles –d *Profilname*

wgetinvunixfiles –f *Profilname*

wgetinvunixfiles –m *Profilname*

wgetinvunixfiles –s *Profilname*

Beschreibung

Mit dem Befehl **wgetinvunixfiles** werden Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, aufgelistet. Folgende Informationen zu einem Profil und dessen Optionen können abgerufen werden:

- Die Verzeichnisse, die bei einem Softwaresuchlauf eingeschlossen oder ausgeschlossen werden.
- Die Dateien oder Dateitypen, die bei einem Softwaresuchlauf eingeschlossen oder ausgeschlossen werden.
- Die MIF-Dateien, die bei einem Suchlauf gelesen werden sollen.
- Die Skripts, die bei einem Suchlauf ausgeführt werden sollen.

Optionen

–d	Gibt eine Liste der Verzeichnisse zurück, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
–f	Gibt eine Liste der Dateien oder Dateitypen zurück, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
–m	Gibt eine Liste der MIF-Dateien zurück, die beim Suchlauf gelesen werden sollen.
–s	Gibt eine Liste der Skripts zurück, die beim Suchlauf auf den Zielen ausgeführt werden sollen.
Profilname	Gibt den Namen des Inventory-Profils an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie diesen Befehl ausführen, ohne Optionen anzugeben, werden alle verfügbaren Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die für das angegebene Inventory-Profil gesammelt werden sollen, aufgelistet.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die durch das Inventory-Profil 'MeinProfil' gesammelt werden sollen, aufgelistet:

```
wgetinvunixfiles @InventoryConfig:MeinProfil
```

Die Ausgabe dieses Befehls ist wie folgt:

```
No Include Directories
Exclude Directories:
        */tmp/
Exclude File Types:
        *.c
        *.txt
No Custom Mif Files
No script to run after scan
```

Weitere Informationen

wgetinvglobal, **wgetinvpcfiles**, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixhw**, **wgetinvunixsw**, **wsetinvunixfiles**

wgetinvunixhw

Mit diesem Befehl werden die für den Hardwaresuchlauf auf einem UNIX-Endpunkt festgelegten Optionen aufgelistet.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvunixhw *Profilname*

wgetinvunixhw –d *Profilname*

wgetinvunixhw –t *Profilname*

wgetinvunixhw–u *Profilname*

Beschreibung

Mit dem Befehl **wgetinvunixhw** werden die Optionen für UNIX-Hardwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden. Folgende Daten zu einem Profil und dessen Optionen können abgerufen werden:

- Ob der Hardwaresuchlauf von dem Profil ausgeführt wird, ob die Suchlaufdaten an das Konfigurations-Repository gesendet werden oder ob beide Vorgänge ausgeführt werden.
- Nach welchen Hardwarekomponenten beim Suchlauf gesucht wird.

Optionen

–d Listet alle Hardwarekomponenten auf, nach denen gesucht werden kann, und gibt an, ob das Inventory-Profil für die Suche nach den jeweiligen Komponenten konfiguriert wurde.

–t Gibt einen Wert zurück, der anzeigt, ob die Tivoli Hardware-Suchfunktion aktiviert wurde. Folgende Werte können zurückgegeben werden:

YES Die Tivoli Hardware-Suchfunktion wurde aktiviert.

NO Die Tivoli Hardware-Suchfunktion wurde nicht aktiviert.

–u Gibt einen Wert zurück, der anzeigt, ob die beim Hardwaresuchlauf gesammelten Daten der UNIX-Systeme an das Konfigurations-Repository gesendet werden. Folgende Werte können zurückgegeben werden:

YES Die Suchlaufdaten werden nach Abschluss des Suchlaufs an das Konfigurations-Repository gesendet.

NO Es werden keine Suchlaufdaten an das Konfigurations-Repository gesendet. Sie werden stattdessen auf dem Endpunkt gespeichert.

Profilname

Gibt den Namen des Inventory-Profils an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über die Optionen für UNIX-Hardwaresuchläufe zurückgegeben, die für das Inventory-Profil 'MeinProfil' konfiguriert wurden:

```
wgetinvunixhw @InventoryConfig:MeinProfil
```

Ausgabe:

```
 Run Tivoli Scanner: YES
Update the hardware scan results in the configuration repository: YES
Data to scan:
        Processor: YES
        Memory: YES
        Operating System: YES
        Storage: YES
        IP Address: YES
        Network Adapter: YES
        Partition: YES
        Pointing Device: YES
        Keyboard: YES
        UNIX System Params: YES
```

Im folgenden Beispiel wird angegeben, ob das Inventory-Profil 'MeinProfil' für die Ausführung der Tivoli Hardware-Suchfunktion konfiguriert wurde:

```
wgetinvunixhw -t @InventoryConfig:MeinProfil
```

Ausgabe:

```
Run Tivoli Scanner: YES
```

Weitere Informationen

wgetinvglobal, **wgetinvpcfiles**, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixsw**, **wsetinvunixhw**

wgetinvunixsw

Mit diesem Befehl werden die Optionen für UNIX-Software- suchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetinvunixsw *Profilname*

wgetinvunixsw –b *Profilname*

wgetinvunixsw –c *Profilname*

wgetinvunixsw –f *Profilname*

wgetinvunixsw –p *Profilname*

wgetinvunixsw –s *Profilname*

wgetinvunixsw –x *Profilname*

Beschreibung

Mit dem Befehl **wgetinvunixsw** werden die Optionen für UNIX-Softwaresuchläufe aufgelistet, die für ein Inventory-Profil konfiguriert wurden. Dieser Befehl gibt Informationen darüber zurück, ob ein Profil für die Ausführung folgender Aktionen konfiguriert wurde:

- Nach grundlegenden Dateiinformationen suchen.
- Für die durchsuchten Dateien Kontrollsummen generieren.
- Angepassten Filter für Suchläufe nach grundlegenden Informationen verwenden.
- Betriebssystem nach Informationen zu installierten Produkten durchsuchen.
- Durchsuchte Dateien mit der Liste der Softwaresignaturen vergleichen.
- Nur ausführbare Dateien durchsuchen.

Optionen

–b Gibt Informationen über Suchläufe nach grundlegenden Dateiinformationen zurück. Folgende Werte können zurückgegeben werden:

SCAN Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Der Endpunkt wird nicht durchsucht. Stattdessen werden die Daten eines früheren Suchlaufs nach grundlegenden Dateiinformationen gesammelt und an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach grundlegenden Dateiinformationen ausgeführt. Es werden keine Daten für grundlegende Dateiinformationen gesammelt.

–c Gibt Informationen zur Konfiguration der Kontrollsummenoptionen zurück. Folgende Werte können zurückgegeben werden:

NONE
Beim Suchlauf werden keine Kontrollsummenwerte der durchsuchten Dateien generiert.

QUICK
Beim Suchlauf wird die schnelle Kontrollsumme der durchsuchten Dateien gesammelt.

FULL Beim Suchlauf wird die vollständige Kontrollsumme der durchsuchten Dateien gesammelt.

MD5 Beim Suchlauf wird die MD5-Kontrollsumme der durchsuchten Dateien gesammelt.

–f Gibt Informationen darüber zurück, ob der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen verwendet wird. Folgende Werte können zurückgegeben werden:

YES Beim Suchlauf wird der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen verwendet.

NO Beim Suchlauf wird der angepasste Filter für Suchläufe nach grundlegenden Dateiinformationen nicht verwendet.

–p Gibt Informationen über Suchläufe im Betriebssystem nach Informationen zu installierten Produkten zurück. Folgende Werte können zurückgegeben werden:

SCAN Das Betriebssystem wird nach Informationen zu installierten Produkten durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Der Endpunkt wird nicht durchsucht. Stattdessen werden die Daten eines früheren Suchlaufs nach Informationen zu installierten Produkten gesammelt und an das Konfigurations-Repository gesendet.

BOTH

Das Betriebssystem wird nach Informationen zu installierten Produkten durchsucht, und die Suchlaufdaten werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach Informationen zu installierten Produkten ausgeführt. Es werden keine Daten über Informationen zu installierten Produkten gesammelt.

–s Vergleicht durchsuchte Dateien mit der Liste der Softwaresignaturen. Folgende Werte können zurückgegeben werden:

SCAN Durchsuchte Dateien werden mit der Liste der Softwaresignaturen verglichen. Daten für übereinstimmende Dateien werden auf dem Endpunkt gespeichert.

UPDATE

Es wird kein Suchlauf nach übereinstimmenden Software signaturen ausgeführt. Stattdessen werden die Daten eines früheren Suchlaufs nach übereinstimmenden Softwaresignaturen gesammelt und an das Konfigurations-Repository gesendet.

BOTH

Durchsuchte Dateien werden mit der Liste der Softwaresignaturen verglichen, und die Daten für übereinstimmende Dateien werden anschließend gesammelt und an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach übereinstimmenden Softwaresignaturen ausgeführt. Es werden keine Daten für übereinstimmende Softwaresignaturen gesammelt.

–x Durchsucht nur ausführbare Dateien. Folgende Werte können zurückgegeben werden:

YES Es werden nur ausführbare Dateien durchsucht.

NO Der Suchlauf ist nicht auf ausführbare Dateien beschränkt.

Profilname Gibt den Namen des Inventory-Profils an, über das Sie Informationen abrufen möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Hinweise

Wenn Sie diesen Befehl ausführen, ohne Optionen anzugeben, werden alle verfügbaren Informationen über die Optionen für UNIX-Softwaresuchläufe zurückgegeben, die für ein bestimmtes Inventory-Profil konfiguriert wurden.

Beispiele

Im folgenden Beispiel werden alle verfügbaren Informationen über die Optionen für UNIX-Softwaresuchläufe zurückgegeben, die für das Inventory-Profil 'MeinProfil' konfiguriert wurden:

```
wgetinvunixsw @InventoryConfig:MeinProfil
```

Von diesem Befehl wird Folgendes zurückgegeben:

```
Scan for and update basic file information: NO
Cyclic Redundancy Check: No CRC
Scan for and update Unix software package information: BOTH
Scan for and update matching software signatures: NO
Apply custom filter to basic file information scan results: NO
Executable files only: YES
```

Im folgenden Beispiel wird die Kontrollsummenoption, die für das Inventory-Profil 'MeinProfil' konfiguriert wurde, zurückgegeben:

```
wgetinvunixsw -c @InventoryConfig:MeinProfil
```

Von diesem Befehl wird Folgendes zurückgegeben:

```
Cyclic Redundancy Check: No CRC
```

Weitere Informationen

wgetinvglobal, **wgetinvpcfiles**, **wgetinvpchw**, **wgetinvpcsw**, **wgetinvunixfiles**, **wgetinvunixhw**, **wsetinvunixsw**

wgetscanstat

Mit diesem Befehl werden Informationen über die aktuellen Inventory-Suchläufe zurückgegeben.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wgetscanstat

wgetscanstat –a [**–s**] [**–f**] [**–p**]

wgetscanstat –i *ID* [**–i** *ID*]... [**–s**] [**–f**] [**–p**]

Beschreibung

Mit dem Befehl **wgetscanstat** werden Informationen über die aktuellen Inventory-Suchläufe zurückgegeben. Wenn Sie diesen Befehl ausführen, ohne Optionen anzugeben, wird eine Liste mit den Suchlauf-IDs der aktuellen Inventory-Suchläufe sowie mit den Namen der Inventory-Profile, die diese Suchläufe eingeleitet haben, zurückgegeben. Mit den verfügbaren Optionen können Sie den Status (erfolgreich, fehlgeschlagen oder anstehend) aller oder bestimmter Suchläufe abrufen.

Optionen

–a Gibt ausführliche Informationen über alle aktuellen Suchläufe zurück, einschließlich der Suchlauf-ID, des Profilnamens, der Startzeit, der abgelaufenen Zeit und der Anzahl der Ziele, auf denen Suchläufe abgeschlossen wurden oder noch anstehen.

–s Gibt eine Liste mit allen Zielen zurück, auf denen Suchläufe erfolgreich abgeschlossen wurden.

–f Gibt eine Liste mit allen Zielen zurück, auf denen Suchläufe fehlgeschlagen sind.

–p Gibt eine Liste mit allen Zielen zurück, auf denen Suchläufe noch anstehen.

–i *ID* Gibt Informationen über den Suchlauf mit der angegebenen Suchlauf-ID zurück. Sie können mehrere Suchlauf-IDs angeben.

Autorisierung

user, **admin**, **senior**, **super** oder **Inventory_view**

Beispiele

Im folgenden Beispiel werden ausführliche Informationen über alle aktuellen Suchläufe zurückgegeben und die fehlgeschlagenen, erfolgreichen und anstehenden Suchläufe aufgelistet.

```
wgetscanstat –a –f –s –p
```

Ausgabe:

```
Scan Id:            144
Profile Name:       Hardware
Start time:         Thu Aug 05 17:18:57 1999
Elapsed time:       0 Days 0 Hours 0 Minutes 11 Seconds
Clients completed:  3
Clients pending:    1
The following clients have successfully completed:
    mckinley
    aztlan–1
The following clients have failed:
    alioth–5
The following clients are still pending:
    suntmp18–3
Scan Id:            145
Profile Name:       Hardware
Start time:         Thu Aug 05 17:18:57 1999
Elapsed time:       0 Days 0 Hours 0 Minutes 11 Seconds
Clients completed:  0
Clients pending:    1
The following clients have successfully completed:
The following clients have failed:
The following clients are still pending:
    suntmp11
```

Weitere Informationen

wcollect, **wcstat**

winvfilter

Mit diesem Befehl werden die Dateien in der Liste der angepassten Filter, die im Konfigurations-Repository gespeichert ist, geändert.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

winvfilter –a –f *Eingabedatei*

winvfilter –r –f *Eingabedatei*

winvfilter –a –n *Filtername*

winvfilter –r –n *Filtername*

winvfilter –a –i

winvfilter –r –i

Beschreibung

Sie können angeben, dass bei Suchläufen nach grundlegenden Dateiinformationen nur die Dateien eingeschlossen werden, die im angepassten Filter aufgeführt sind. Mit dem Befehl **winvfilter** können Sie in der Liste der angepassten Filter Dateinamen hinzufügen oder löschen. Die Liste der angepassten Filter ist im Konfigurations-Repository gespeichert. Mit der Option **–n** können Sie jeweils einen Dateinamen und mit der Option **–f** gleichzeitig mehrere Dateinamen hinzufügen. Weitere Informationen zum angepassten Filter finden Sie in der Onlinehilfe zu Tivoli Inventory.

Anmerkung: Auf UNIX-Systemen muss bei Dateinamen die Groß-/Kleinschreibung beachtet werden.

Der angepasste Filter wird aktiviert, indem Sie den Befehl **wsetinvpcsw** oder **wsetunixsw** mit der Option **–f** verwenden oder im Fenster 'Konfiguration für Softwaresuchlauf' die Option 'Angepassten Filter übernehmen' auswählen.

Optionen

–a Fügt der Liste der angepassten Filter die angegebene Datei hinzu.

–f *Eingabedatei*
Gibt den Namen einer Textdatei an, die eine Liste mit Dateinamen enthält, die dem angepassten Filter hinzugefügt oder daraus gelöscht werden sollen. In der Eingabedatei muss jeder Dateiname in einer separaten Zeile stehen.

–i Gibt Filterdaten an, die aus der Standardeingabeeinheit gelesen werden sollen. Sie geben die Daten mit Hilfe einer Piping-Eingabe über einen Befehl ein.

–n *Filtername* Gibt einen Dateinamen an, der dem angepassten Filter hinzugefügt oder daraus gelöscht werden soll.

–r Entfernt die angegebene Datei oder Liste mit Dateien aus dem angepassten Filter.

Autorisierung

RIM_update, user, admin, senior oder **super**

Beispiele

Im folgenden Beispiel wird die Liste der Dateinamen in der Datei 'filterliste.txt' dem angepassten Filter hinzugefügt.

```
winvfilter -a -f filterliste.txt
```

Im folgenden Beispiel wird der Dateiname 'MeineAnw.exe' aus dem angepassten Filter entfernt.

```
winvfilter -r -n MeineAnw.exe
```

Im folgenden Beispiel werden die Dateinamen aus der Softwaresignaturentabelle (SWARE_SIG) in die Liste der angepassten Filter geladen. Bei diesem Beispiel wird davon ausgegangen, dass ein Verwaltungssystem für relationale DB2-Datenbanken (DB2 RDBMS) verwendet wird, dass eine DB2-Umgebung die Quelle ist und eine Verbindung zur Datenbank hergestellt wurde:

```
db2 "select sware_name from sware_sig" | winvfilter -a -i
```

Weitere Informationen

winvsig, **wsetinvpcsw**, **wsetinvunixsw**

winvrmnode

Mit diesem Befehl werden alle durch Suchläufe erfassten Informationen zu einem bestimmten Endpunkt aus dem Konfigurations-Repository entfernt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

winvrmnode –i *Computersystem_ID ...*

winvrmnode –l *TME_Objektbezeichnung ...*

winvrmnode –o *TME_Objekt_ID ...*

Beschreibung

Mit dem Befehl **winvrmnode** können Sie Hardware- und Softwareinformationen zu einem bestimmten Endpunkt aus dem Konfigurations-Repository von Tivoli Inventory entfernen. Mit diesem Befehl werden jedoch keine Daten aus den Protokolltabellen und angepassten Tabellen entfernt.

Sie können den gewünschten Endpunkt festlegen, indem Sie den Namen (*TME_Objektbezeichnung*), die Objekt-ID (*TME_Objekt_ID*) oder die Computersystem-ID (*Computersystem_ID*) des Systems angeben. Wenn mehrere Endpunkte mit demselben Namen vorhanden sind, werden bei Angabe der Option *TME_Objektbezeichnung* die Tivoli Inventory-Informationen für alle Endpunkte mit diesem Namen entfernt, während bei Angabe der Option *TME_Objekt_ID* oder *Computersystem_ID* nur die Informationen für den angegebenen Endpunkt entfernt werden. Bei Angabe einer Namensliste können Sie die Daten mehrerer Endpunkte gleichzeitig entfernen.

Anmerkung: Bei der Objekt-ID eines Systems handelt es sich um den für die Variable 'TME_OBJECT_ID' angegebenen Wert. Die Objekt-ID eines Systems können Sie ermitteln, indem Sie die Abfrage 'INVENTORY_HWARE' unter Angabe der Spalten 'TME_OBJECT_ID' und

’TME_OBJECT_LABEL’ ausführen. Als Ergebnis werden die Systemnamen mit den entsprechenden Objekt-IDs zurückgegeben. Bei der Computersystem-ID eines Systems handelt es sich um den für die Variable ’COMPUTER_SYS_ID’ angegebenen Wert. Die Computersystem-ID eines Systems können Sie ermitteln, indem Sie die Abfrage ’INVENTORY_HWARE’ ausführen.

Optionen

–i *Computersystem_ID*
: Entfernt alle Hardware- und Softwareinformationen zu dem Endpunkt mit der angegebenen Computersystem-ID aus dem Konfigurations-Repository.

–l *TME_Objektbezeichnung*
: Entfernt alle Hardware- und Softwareinformationen zu dem Endpunkt mit dem angegebenen Namen aus dem Konfigurations-Repository. Wenn mehrere Instanzen derselben Bezeichnung vorhanden sind, werden mit dieser Option die Informationen für alle Instanzen entfernt.

–o *TME_Objekt_ID*
: Entfernt alle Hardware- und Softwareinformationen zu dem Endpunkt mit der angegebenen Objekt-ID aus dem Konfigurations-Repository.

Autorisierung

super

Hinweise

Informationen zur Verwendung des Befehls **winvrmnode**, um Suchlaufdaten aus angepassten Tabellen zu entfernen, finden Sie unter „Benutzerspezifische Tabelle im Konfigurations-Repository hinzufügen” auf Seite 141.

Beispiele

Mit folgendem Befehl werden alle Hardware- und Softwareinformationen zu den Endpunkten ’Nil’ und ’Tigris’ entfernt:

```
winvrmnode -l Nil Tigris
```

winvsig

Mit diesem Befehl wird die Liste der Softwaresignaturen im Konfigurations-Repository geändert.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

winvsig –a –f *Dateiname*

winvsig –a –i

winvsig –a –n *Name* **–s** *Größe* **–d** *Beschreibung* **–v** *Version* [**–q** *schnelle Kontrollsumme* | **–c** *vollständige Kontrollsumme* | **–m** *MD5-Kontrollsumme*]

winvsig –r –f *Dateiname*

winvsig –r –i

winvsig –r –n *Name* **–s** *Größe*

Beschreibung

Mit dem Befehl **winvsig** werden im Konfigurations-Repository Softwaresignaturen hinzugefügt, geändert oder entfernt. Bei einem Suchlauf nach übereinstimmenden Softwaresignaturen gleicht Tivoli Inventory die auf dem Endpunkt gespeicherten Dateien mit den Softwaresignaturdaten ab. Wenn die durchsuchte Datei in den Werten für Dateiname und Größe mit einer Softwaresignatur übereinstimmt, sendet Tivoli Inventory die Softwaresignaturdaten für die übereinstimmende Datei an das Konfigurations-Repository.

Anmerkung: Auf UNIX-Systemen muss bei der Angabe des Wertes für 'Dateiname' die Groß-/Kleinschreibung beachtet werden.

Optionen

–a Fügt dem Konfigurations-Repository eine oder mehrere Softwaresignaturen hinzu und ändert vorhandene Softwaresignaturen.

–c *vollständige Kontrollsumme*
Gibt den vollständigen CRC32-Kontrollsummenwert (CRC32_CHECKSUM) der Datei an, die Sie mit der Option **–n** angeben. Hierbei handelt es sich um einen aus acht Zeichen bestehenden hexadezimalen Wert; alphabetische Zeichen müssen in Kleinbuchstaben angegeben werden.

–d *Beschreibung*
Gibt einen Namen bzw. eine Beschreibung für die Software an, die der Datei zugeordnet ist, die Sie mit der Option **–n** angeben. Wenn die Beschreibung Leerzeichen enthält, muss sie in Anführungszeichen (″ ″) gesetzt werden. Diese Angabe wird in der grafischen Benutzerschnittstelle von Tivoli Inventory im Fenster 'Signaturen' in der Spalte 'Softwarename' angezeigt.

–f *Dateiname* Gibt den Namen einer Datei an, die die Daten für eine oder mehrere Softwaresignaturen enthält. Die Softwaresignaturdaten in der Datei müssen in folgendem Format angegebenen werden:

```
<I>,Name,Größe,Beschreibung,Version,schnelle
Kontrollsumme,vollständige Kontrollsumme,
MD5-Kontrollsumme
```

Die Werte für *schnelle Kontrollsumme, vollständige Kontrollsumme* und *MD5-Kontrollsumme* sind optional. Sie können einen oder mehrere dieser Werte angeben. Die Werte müssen jedoch in der angezeigten Reihenfolge angegeben werden. Außerdem ist es besonders wichtig, falls Sie einen Wert überspringen möchten, dass Sie dafür einen Platzhalter in Form von zwei doppelten Anführungszeichen (″″) einsetzen. Wenn Sie beispielsweise einen Wert für *voll-*

ständige Kontrollsumme und *MD5-Kontrollsumme*, nicht jedoch für *schnelle Kontrollsumme* angeben möchten, geben Sie folgende Daten ein:

```
<I>,Name,Größe,Beschreibung,Version,"", vollständige
Kontrollsumme,MD5-Kontrollsumme
```

–i Gibt Softwaresignaturdaten an, die aus der Standardeingabeeinheit gelesen werden sollen. Sie geben die Daten mit Hilfe einer Piping-Eingabe über einen Befehl ein.

–m *MD5-Kontrollsumme*
Gibt den MD5-Kontrollsummenwert (MD5_CHECKSUM) der Datei an, die Sie mit der Option **–n** angeben. Hierbei handelt es sich um einen aus 32 Zeichen bestehenden hexadezimalen Wert; alphabetische Zeichen müssen in Kleinbuchstaben angegeben werden.

–n *Name* Gibt den Namen der Datei an, über die das Softwareprodukt identifiziert werden soll. Dies ist normalerweise die primäre ausführbare Datei für ein Softwareprodukt, z. B. notepad.exe. Die Kombination aus Dateiname, der mit der Option **–n** angegeben wird, und Dateigröße, die mit der Option **–s** angegeben wird, muss in der Liste der Softwaresignaturen eindeutig sein.

Sie müssen den Dateinamen so angeben, wie er definiert ist, einschließlich Groß-/Kleinschreibung. Wenn sich der Dateiname in verschiedenen Betriebssystemen unterscheidet, müssen Sie unter Umständen für jedes Betriebssystem eine eigene Softwaresignatur hinzufügen. Beispielsweise handelt es sich bei 'SAMPLE.EXE' und 'sample.exe' um zwei verschiedene Dateien.

–q *schnelle Kontrollsumme*
Gibt den schnellen Kontrollsummenwert (QUICK-_CHECKSUM) der Datei an, die Sie mit der Option **–n** angeben. Hierbei handelt es sich um einen aus acht Zeichen bestehenden hexadezimalen Wert; alphabetische Zeichen müssen in Kleinbuchstaben angegeben werden.

–r Entfernt eine oder mehrere Softwaresignaturen aus dem Konfigurations-Repository.

–s *Größe* Gibt die Größe der Datei in Byte an, die Sie mit der Option **–n** angeben. Die Kombination aus Dateiname, der mit der Option **–n** angegeben wird, und Dateigröße, die mit der Option **–s** angegeben wird, muss in der Liste der Softwaresignaturen eindeutig sein.

Anmerkung: Hier ist die Angabe eines Wertes größer Null erforderlich. Wenn Sie für diese Option den Wert ’0’ angeben, schlägt der Befehl **winvsig** fehl.

–v *Version* Gibt die Version des Softwarepakets an, das der Datei zugeordnet ist, die Sie mit der Option **–n** angeben.

Autorisierung

RIM_update, user, admin, senior oder **super**

Hinweise

Alle hexadezimalen Werte, die Sie in diesem Befehl angeben, müssen durch vorangestellte Nullen auf acht Ziffern aufgefüllt werden:

Falsch:

```
ff1234
```

Richtig:

```
00ff1234
```

Beispiele

Mit dem folgenden Befehl wird dem Konfigurations-Repository eine Softwaresignatur für die Version 1.0 des Softwarepakets Windows NT WordPad hinzugefügt.

```
winvsig –a –n wordpad.exe –s 204800 –d "WordPad" –v 1.0
```

Im folgenden Beispiel werden die in der Datei 'Dateien.txt' aufgeführten Softwaresignaturdaten aus dem Konfigurations-Repository entfernt.

```
winvsig –r –f Dateien.txt
```

Im folgenden Beispiel wird die Version einer vorhandenen Softwaresignatur auf '1.2' aktualisiert und der schnelle Kontrollsummenwert hinzugefügt.

```
winvsig –a –n sample.exe -s 2048 -v 1.2 -q fd123c34
```

Weitere Informationen

winvfilter, **wqueryinv**

wqueryinv

Mit diesem Befehl werden aus dem Konfigurations-Repository Daten zu einem Client in der Tivoli-Umgebung abgefragt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wqueryinv –d *Begrenzer* {**–o** *Endpunkt_OID* | *Endpunktname*}

wqueryinv [**–h** *Host-Name*] **–f** *Dateiname* {**–o** *Endpunkt_OID* | *Endpunktname*}

wqueryinv –n {**–o** *Endpunkt_OID* | *Endpunktname*}

wqueryinv {**–s** | **–q @***Abfrage*} {**–o** *Endpunkt_OID* | *Endpunktname*}

Beschreibung

Mit dem Befehl **wqueryinv** werden Inventory-Suchlaufdaten für einen Endpunkt in der Tivoli-Umgebung abgerufen. Sie können diesen Befehl für Abfragen im Konfigurations-Repository verwenden, in denen die Spalten 'TME_OBJECT_LABEL' und 'TME_OBJECT_ID' für alle Endpunkte in der Tivoli-Umgebung enthalten sind. Die Abfrageergebnisse können entweder an die Standardausgabeeinheit gesendet oder in einer Datei gespeichert werden. Wenn Sie keine bestimmte Abfrage angeben, wird mit diesem Befehl die Abfrage 'INVENTORY_HWARE' für den angegebenen Client ausgeführt.

Sie können mit dem Befehl **wqueryinv** eine Abfrage zu einem Endpunkt veranlassen, indem Sie dessen Objekt-ID oder Host-Namen angeben.

Optionen

–d *Begrenzer* Gibt den Begrenzer an, der in der Datei mit den Abfrageergebnissen verwendet werden soll. Das Begrenzungszeichen muss in Anführungszeichen (″ ″) gesetzt werden. Erfolgt keine Angabe, werden Kommas verwendet.

–f *Dateiname* Gibt den Namen und Pfad der Datei an, in der die Abfrageergebnisse gespeichert werden. Sie müssen sowohl ein Verzeichnis als auch einen Dateinamen angeben. Wenn Sie die Option **–f** nicht verwenden, werden die Abfrageergebnisse an die Standardausgabeeinheit gesendet.

–h *Host-Name* Gibt den verwalteten Knoten an, auf dem die Datei mit den Abfrageergebnissen gespeichert werden soll. Wenn Sie die Option **–h** nicht verwenden, werden die Abfrageergebnisse auf dem lokalen System gespeichert.

–n Gibt an, dass die Datei mit den Abfrageergebnissen keine Kopfzeilen enthalten soll.

–o *Endpunkt_OID*
Gibt die Tivoli-Objekt-ID des Endpunkts an, für den die Abfrage gestellt wird.

–q @*Abfrage* Gibt die Abfrage an, die ausgeführt werden soll.

–s Führt die Abfrage ’INVENTORY_SWARE’ aus, mit der Softwareinformationen zum angegebenen Client zurückgegeben werden. Wenn Sie die Option **–s** oder **–q** nicht verwenden, wird die Abfrage ’INVENTORY_HWARE’ ausgeführt.

Endpunkt-Name
Gibt den Host-Namen des Endpunkts an, für den die Abfrage ausgeführt wird.

Autorisierung

RIM_view, **Query_view**, **Query_execute**, **Query_edit**, **senior** oder **super**

Beispiele

Im folgenden Beispiel wird die Abfrage 'HDISK_QUERY' für den Endpunkte 'alister' ausgeführt und die Ausgabe an die Datei '/tmp/inv.out' auf dem System 'mckinley' gesendet.

```
wqueryinv –n –d ";" –h mckinley –f /tmp/inv.out \
–q @HDISK_QUERY alister
```

Weitere Informationen

wrunquery (im Handbuch *Tivoli Management Framework Reference Manual*)

wsetinvdh

Mit diesem Befehl werden die Einstellungen für den Inventory Data Handler geändert.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvdh [**–d** *Statusverzeichnis*]

wsetinvdh [**–n** *alle n Minuten bündeln*]

wsetinvdh [**–q** *alle n Ziele bündeln*]

wsetinvdh [**–r** *Max. Anzahl RDBMS-Einträge*]

wsetinvdh [**–s** {**YES** | **NO**}]

wsetinvdh [**–t** *Verzögerungszeit RDBMS-Wiederholung*]

Beschreibung

Mit dem Befehl **wsetinvdh** können Sie Änderungen am Inventory Data Handler vornehmen, den Sie mit dem Befehl **wcrtinvdh** erstellt haben. Die Optionen sind mit denen des Befehls **wcrtinvdh** identisch, mit der einzigen Ausnahme, dass Sie hier keinen verwalteten Knoten angeben. Mit dem Befehl **wsetinvdh** werden die Änderungen automatisch an der durch '@InvDataHandler:inv_data_handler' bezeichneten Instanz des Inventory Data Handler in der Tivoli-Objektdatenbank vorgenommen.

Optionen

–d *Statusverzeichnis*

Gibt an, in welchem Verzeichnis der Inventory Data Handler Statusinformationen speichert, die im Falle eines Systemausfalls wiederhergestellt werden können. Standardmäßig ist dies das Verzeichnis '$DBDIR/inventory/stat_dir' auf dem verwalteten Knoten, auf dem sich der Inventory Data Handler befindet.

–n *alle n Minuten bündeln*
Gibt das Intervall an, in dem Nachrichten über den Abschluss von Suchläufen gesendet werden sollen (wenn der Option **–n** des Befehls **wsetinvglobal** der Wert **BUNDLE** zugeordnet wurde). Wenn Sie diesen Wert festlegen, sendet Tivoli Inventory eine Nachricht mit einer Liste der Ziele, auf denen in der angegebenen Zeitspanne Suchläufe abgeschlossen wurden. Der Standardwert ist 10 Minuten.

Wenn in der angegebenen Zeitspanne kein Suchlauf abgeschlossen wurde, wird auch keine Nachricht gesendet.

Wenn Sie für diese Option den Wert '0' angeben, erfolgt die Benachrichtigung entsprechend dem Wert, der der Option **–q** des Befehls **wcrtinvdh** oder des Befehls **wsetinvdh** zugeordnet wurde.

–q *alle n Ziele bündeln*
Gibt die maximale Anzahl von Zielen in einem Bündel an. Ein Bündel ist eine Gruppe von Zielen, deren Statusinformationen zu demselben Zeitpunkt gesendet werden. Der Status gibt für jedes Ziel eines bestimmten Inventory-Profils an, ob ein Suchlauf erfolgreich war oder fehlgeschlagen ist.

Der Standardwert ist 10 Ziele.

Mit dieser Option in Verbindung mit der Option **–n**, die im Befehl **wsetinvglobal** angegeben wird, wird festgelegt, wie Statusinformationen in Bündeln gesendet werden sollen.

Anmerkungen:

1. Wenn Sie sowohl der Option **–q** *alle n Zielen bündeln* als auch der Option **–n** *alle n Minuten bündeln* den Wert '0' zuweisen, findet keine Bündelung statt, und Sie erhalten erst dann eine Nachricht, wenn die Suchläufe auf allen Zielen abgeschlossen wurden.

2. Wenn Sie sowohl der Option **–q** *alle n Ziele bündeln* als auch der Option **–n** *alle n Minuten bündeln* einen positiven Wert zuordnen, findet jedes Mal eine Bündelung statt, wenn einer der angegebenen Werte (Anzahl der Ziele oder Minuten) erreicht ist.

–r *Max. Anzahl RDBMS-Einträge*

Gibt an, wie oft der Inventory Data Handler versucht, Daten auf den RDBMS-Server zu schreiben. Wenn die maximale Anzahl der Wiederholungen erreicht ist, wird eine Fehlernachricht gesendet.

Der Standardwert ist 5.

–s *Option*

Gibt an, ob der Inventory Data Handler Statusinformationen speichern soll, die im Falle eines Systemausfalls wiederhergestellt werden können. Sie können eine der folgenden Optionen verwenden:

YES Statusinformationen werden gespeichert. Dies ist die Standardoption.

NO Statusinformationen werden nicht gespeichert.

–t *Verzögerungszeit RDBMS-Wiederholung*

Gibt einen Wert an, auf dessen Grundlage das Zeitlimit zwischen wiederholten Versuchen, Daten in ein RIM-Objekt zu schreiben, berechnet wird. Das Zeitlimit wird nach dem Algorithmus *Zeitlimit*Wiederholungszähler* berechnet. Beispiel: Bei der ersten Wiederholung mit einem Zeitlimitwert von 30 Sekunden wird der Zeitgeber durch den Algorithmus auf 30 * 1, also 30 Sekunden gesetzt. Bei der zweiten Wiederholung wird der Zeitgeber auf 30 * 2, also 1 Minute gesetzt.

Der Standardwert ist 30 Sekunden.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Im folgenden Beispiel wird die maximale Anzahl von Zielen in einem Hinweisbündel auf 9 und die Verzögerungszeit für Schreibvorgänge im RIM-Objekt auf 20 Sekunden gesetzt.

```
wsetinvdh –q 9 –t 20
```

Weitere Informationen

wcollect, **wcrtinvdh**, **wgetinvdh**, **wsetinvglobal**

wsetinvglobal

Mit diesem Befehl werden die allgemeinen Merkmale für ein Inventory-Profil festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvglobal –d {**CONFIG** | **ALL**} *Profilname*

wsetinvglobal –e *Sekunden Profilname*

wsetinvglobal –f *Protokolldateiname Profilname*

wsetinvglobal –h *Host-Protokolldatei Profilname*

wsetinvglobal –l {**OFF** | *Optionsliste*} *Profilname*

wsetinvglobal –m {**Y** | **N**} *Profilname*

wsetinvglobal –n {**IMMEDIATE** | **BUNDLE** | **DONE**} *Profilname*

wsetinvglobal –t {**SUCCESS** | **FAIL** | **ALL**} *Profilname*

wsetinvglobal –u {**REPLACE** | **DIFFS**} *Profilname*

wsetinvglobal –w {**Y** | **N**} *Profilname*

Beschreibung

Mit dem Befehl **wsetinvglobal** können Sie die allgemeinen Merkmale für ein Inventory-Profil festlegen. Folgende Merkmale können festgelegt werden:

- Verteilungsoptionen - Sie können angeben, ob das Inventory-Profil nur die Konfigurationsdatei verteilt oder ob es die Konfigurationsdatei verteilt und anschließend den Suchlauf ausführt.
- Datenoptionen - Sie können angeben, ob beim Suchlauf nur neue oder geänderte Daten an das Konfigurations-Repository gesendet

werden oder ob vorhandene Daten im Konfigurations-Repository durch Daten, die beim Suchlauf gesammelt wurden, ersetzt werden.

- Statusoptionen - Sie können angeben, wohin und unter welchen Bedingungen Hinweise zum Suchlaufstatus gesendet werden. Außerdem können Sie den Namen und Pfad der Statusprotokolldatei und den verwalteten Knoten, auf dem sie gespeichert wird, angeben.

Optionen

–d *Option*

Gibt die Verteilungsoptionen an. Sie können eine der folgenden Optionen verwenden:

CONFIG
: Nur die Konfigurationsdatei verteilen.

Alle
: Konfigurationsdatei verteilen und den Suchlauf ausführen. Dies ist die Standardoption.

–e *Sekunden*

Gibt die Zeit in Sekunden an, die dem Profil für den Suchlauf auf einem Endpunkt eingeräumt werden, bevor es abläuft. Der Standardwert beträgt 1800 Sekunden.

–f *Protokolldateiname*

Gibt die Position und den Namen einer Datei an, in der die Statusinformationen für Tivoli Inventory gespeichert werden sollen. Dabei muss ein vollständiger Pfad angegeben werden, relative Pfadangaben sind nicht zulässig. Wenn Sie eine leere Zeichenfolge angeben (″″) wird das Protokoll in eine Datei mit dem Namen *$TMPDIR*/inv_scan_*nn*.log geschrieben, wobei *nn* für die Suchlauf-ID und *$TMPDIR* für ein temporäres Verzeichnis steht. Das temporäre Verzeichnis befindet sich normalerweise in einem der folgenden Pfade:

- Unter UNIX: ’/tmp’, ’/usr/tmp’ oder ’/var/tmp’
- Unter Windows NT: ’c:\temp’

–h *Host-Protokolldatei*

Gibt den Namen eines verwalteten Knotens an, auf dem die Statusinformationen für Tivoli Inventory gespeichert werden sollen. Wenn Sie eine leere Zeichenfolge (″″) angeben, wird die Host-Protokolldatei auf dem verwalteten Knoten gespeichert, auf dem der Inventory Data Handler erstellt wurde.

–l {OFF | *Optionsliste***}**

Gibt eine Position oder mehrere Adressen an, an die Hinweise gesendet werden sollen. Sie können eine oder mehrere der folgenden Optionen verwenden:

NOTICE_GROUP

Statusinformationen werden an die Tivoli Inventory-Hinweisgruppe gesendet.

LOG_FILE

Statusinformationen werden in der Protokolldatei gespeichert, die mit Hilfe der Optionen **–f** und **–h** angegeben wird.

TEC Sendet Informationen an die TEC-Konsole (Tivoli Enterprise Console)..

OFF Es werden keine Statusinformationen gesendet.

Der Standardwert ist 'NOTICE_GROUP'. Sie können mehrere Adressen angeben, indem Sie mehrere Optionen verwenden. Die Optionen müssen durch ein Komma (,) voneinander getrennt werden.

Anmerkung: Wenn beim Schreiben der Protokolldatei in Tivoli Inventory ein Fehler auftritt, wird eine Fehlernachricht an die Tivoli Inventory-Hinweisgruppe gesendet, und sämtliche Statusinformationen werden ausschließlich an die Tivoli Inventory-Hinweisgruppe gesendet.

–m Gibt an, ob eine Verteilung an einen mobilen Endpunkt bzw. Endpunkt für standortunabhängigen Zugriff (wie z. h. einen Laptop) ausgeblendet sein soll. Ausgeblendete Verteilungen werden automatisch an den Endpunkt für standortunab-

hängigen Zugriff verteilt und werden für den Benutzer dieses Endpunkts nicht sichtbar ausgeführt. Verteilungen, die nicht ausgeblendet wurden, sind für den Benutzer mit standortunabhängigem Zugriff sichtbar, und ihre Ausführung kann von ihm verzögert werden. Sie können eine der folgenden Optionen auswählen:

Y Die Verteilung wird ausgeblendet.

N Die Verteilung wird nicht ausgeblendet.

–n *Option*

Gibt an, wann ein Hinweis auf den Abschluss eines Suchlaufs auf den jeweiligen Zielen gesendet wird. Sie können eine der folgenden Optionen verwenden:

IMMEDIATE

Der Hinweis wird unmittelbar nach Abschluss des Suchlaufs auf jedem einzelnen Ziel gesendet.

BUNDLE

Der Hinweis wird in regelmäßigen Abständen gemäß den Einstellungen für den Inventory Data Handler gesendet. Weitere Informationen zum Bündeln finden Sie unter den Befehlen **wcrtinvdh**, **wgetinvdh** und **wsetinvdh**.

DONE

Der Hinweis wird erst gesendet, nachdem die Suchläufe auf allen Zielen abgeschlossen wurden.

Der Standardwert ist 'DONE'.

–t *Option*

Gibt die Umstände an, unter denen ein Hinweis für das angegebene Profil gesendet wird. Sie können eine der folgenden Optionen auswählen:

NONE

Es wird kein Hinweis gesendet.

SUCCESS

Der Hinweis wird für Ziele gesendet, auf denen der Suchlauf erfolgreich abgeschlossen wurde.

FAIL Der Hinweis wird für Ziele gesendet, auf denen der Suchlauf fehlgeschlagen ist.

All Der Hinweis wird für alle Ziele gesendet.

Der Standardwert ist 'ALL'.

–u *Option*

Gibt die Datenoptionen an. Sie können eine der folgenden Optionen verwenden:

REPLACE

Ersetzt vorhandene Daten im Konfigurations-Repository durch Daten, die beim Suchlauf gesammelt wurden.

DIFFS

Sendet nur die Daten an das Konfigurations-Repository, die seit dem letzten Suchlauf hinzugefügt oder geändert wurden. Daten aus vorherigen Suchläufen, die im aktuellen Suchlauf nicht mehr vorhanden sind, werden aus dem Konfigurations-Repository gelöscht. Die Änderungen werden nur aufgezeichnet, wenn Protokolltabellen verwendet werden. Dies ist die Standardoption.

–w Gibt an, ob Systeme mit einer WoL-fähigen Netzwerkkarte (Wake-on-LAN) bei der Profilverteilung automatisch gestartet werden. Sie können eine der folgenden Optionen auswählen:

Y Das System wird bei der Profilverteilung gestartet.

N Das System wird bei der Profilverteilung nicht gestartet.

Profilname

Gibt das Profil an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig-:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Im folgenden Beispiel wird das Inventory-Profil 'SW_Profile' so konfiguriert, dass vorhandene Daten im Konfigurations-Repository durch Daten, die beim Suchlauf gesammelt wurden, ersetzt werden und die Statusinformationen jedes Mal gesendet werden, wenn der Vorgang auf einem Ziel abgeschlossen wurde.

```
wsetinvglobal -u REPLACE -n IMMEDIATE @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird das Profil 'SW_Profile' so konfiguriert, dass die Hinweise an die Protokolldatei, die Tivoli Inventory-Hinweisgruppe und die TEC-Konsole gesendet werden.

```
wsetinvglobal -l LOG_FILE,NOTICE_GROUP,TEC \
@InventoryConfig:SW_Profile
```

Weitere Informationen

wgetinvglobal, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvpcsw**, **wsetinvunixfiles**, **wsetinvunixhw**, **wsetinvunixsw**

wsetinvpcfiles

Mit diesem Befehl werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, angegeben.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvpcfiles –e {**INCLUDE** | **EXCLUDE**} *Profilname*

wsetinvpcfiles –f {**+** | –}*Dateityp Profilname*

wsetinvpcfiles –m{**+** | **–**}*MIF-PfadProfilname*

wsetinvpcfiles –s *SkriptpfadProfilname*

wsetinvpcfiles –t {**INCLUDE** | **EXCLUDE**} **–d** {**+** | **–**}*VerzeichnisProfilname*

Beschreibung

Mit dem Befehl **wsetinvpcfiles** werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf PC-Endpunkten gesammelt werden sollen, angegeben. Mit diesem Befehl können Sie folgende Optionen festlegen:

- Die Dateien oder Dateitypen, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
- Die Verzeichnisse, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
- Die Skripts, die nach Abschluss des Suchlaufs bei der Profilverteilung ausgeführt werden sollen.
- Die benutzerdefinierten MIF-Dateien, die nach Abschluss des Suchlaufs bei der Profilverteilung gesammelt werden sollen.

Optionen

–d *Option Verzeichnis*

Gibt ein Verzeichnis an, das in der Liste der Verzeichnisse, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen, hinzugefügt oder entfernt werden soll. Standardmäßig werden die Verzeichnisse '*/tmp' und '*/temp' beim Suchlauf ausgeschlossen. Um anzuzeigen, ob das Verzeichnis beim Suchlauf eingeschlossen oder ausgeschlossen werden soll, müssen Sie dieser Option die Option **–t** voranstellen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Das Verzeichnis wird der Liste hinzugefügt.

– Das Verzeichnis wird aus der Liste entfernt.

Bei Angabe eines Verzeichnisses, das in einen Suchlauf *eingeschlossen* werden soll, müssen die folgenden Hinweise befolgt werden:

- Sie können keine Platzhalterzeichen verwenden. Alle Zeichen werden als Literale behandelt, auch die Zeichen Stern (*) und Fragezeichen (?).
- Wenn Sie kein Laufwerk angeben (z. B. 'c:/' oder 'sys:/'), durchsucht Tivoli Inventory das angegebene Verzeichnis auf allen Laufwerken.
- Alle Verzeichnisse werden als absolute Pfadangaben behandelt. Bei der Angabe von 'abc' wird 'c:/abc' durchsucht, nicht aber 'c:/temp/abc'.

Bei Angabe eines Verzeichnisses, das von einem Suchlauf *ausgeschlossen* werden soll, müssen die folgenden Hinweise beachtet werden:

- Sie können am Anfang oder Ende eines Pfadnamens einen Stern als Platzhalterzeichen verwenden, z. B. '*abc', '*/abc' oder 'abc/def*'.

- Sie können auch nach einer Laufwerkbezeichnung einen Stern als Platzhalterzeichen verwenden, z. B. 'c:*/abc' oder 'sys:*/abc'.
- Innerhalb eines Pfadnamens können Sie kein Platzhalterzeichen verwenden, z. B. '/*/temp', 'c:/*/temp', 'te*mp' oder '/user*/temp'). Alle Zeichen innerhalb von Pfadangaben werden als Literale behandelt.
- Sie können anstelle eines Laufwerkbuchstabens kein Platzhalterzeichen verwenden.
- Verzeichnisse, die einen Stern als Platzhalterzeichen enthalten, müssen in Anführungszeichen (″ ″) gesetzt werden.

–e *Option*

Gibt an, ob die Liste der Dateien oder Dateitypen, die mit der Option **–f** erstellt wurde, beim Suchlauf eingeschlossen oder ausgeschlossen werden soll. Sie können eine der folgenden Optionen verwenden:

INCLUDE

Die Dateien oder Dateitypen werden beim Suchlauf eingeschlossen. Dies ist die Standardoption.

EXCLUDE

Die Dateien oder Dateitypen werden beim Suchlauf ausgeschlossen.

–f *Option Dateityp*

Gibt eine Datei oder einen Dateityp an, die oder der in der Liste der Dateien oder Dateitypen, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen, hinzugefügt oder entfernt werden soll. Sie können eine Datei mit Hilfe des Dateinamens bzw. einen Dateityp mit Hilfe eines Sterns (*) als Platzhalterzeichen angeben. Sie können den Stern an den Anfang, an das Ende oder an beide Positionen der Zeichenfolge stellen (Beispiele: '*.bat', 'abc.*', '*abc.bat', 'abc.ba*', '*abc', 'abc*', '*abc*'). Innerhalb der Zeichenfolge können Sie jedoch keinen Stern verwenden (Beispiele: 'ab*c', 'ab*c.bat', 'abc.*at'). Dateitypen müssen in Anfüh-

rungszeichen (″ ″) gesetzt werden. Sie können die Option **–f** mehrmals verwenden, um mehrere Dateien oder Dateitypen hinzuzufügen oder zu entfernen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Die Datei oder der Dateityp wird der Liste hinzugefügt.

– Die Datei oder der Dateityp wird aus der Liste entfernt.

–m *Option MIF-Pfad*

Gibt den Pfad und Namen einer MIF-Datei an, die in der Liste der MIF-Dateien, die bei der Profilverteilung gesammelt werden sollen, hinzugefügt oder entfernt werden soll. Für die Option *MIF-Pfad* muss der vollständige Pfad und Name der MIF-Datei angegeben werden. Sie können die Option **–m** mehrmals verwenden, um mehrere MIF-Dateien hinzuzufügen oder zu entfernen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Die MIF-Datei wird der Liste hinzugefügt.

– Die MIF-Datei wird aus der Liste entfernt.

–s *Skriptpfad*

Gibt den Pfad und Namen eines Skripts an, das bei der Profilverteilung auf einem Endpunkt ausgeführt werden soll. Das Skript wird nach Abschluss des Suchlaufs, jedoch vor dem Lesen der MIF-Dateien ausgeführt. Es können nur Textdateien angegeben werden. Die Angabe von Binärdateien ist nicht zulässig.

Anmerkungen:

1. Wenn Sie ein Skript editieren, nachdem Sie es hinzugefügt haben, muss es anschließend erneut mit der Option **–s** hinzugefügt werden. Wenn Sie diesen Schritt nicht ausführen, wird das editierte Skript nicht ausgeführt.
2. Wenn ein Skript entfernt werden soll, geben Sie Anführungszeichen ohne Zeichen dazwischen (″″) ein.

–t *Option*

Gibt an, ob das Verzeichnis, das mit der im Anschluss an diese Option eingegebenen Option **–d** angegeben wird, bei Softwaresuchläufen auf PC-Systemen eingeschlossen oder ausgeschlossen werden soll. Sie können eine der folgenden Optionen verwenden:

INCLUDE

Das angegebene Verzeichnis wird beim Suchlauf eingeschlossen.

EXCLUDE

Das angegebene Verzeichnis wird beim Suchlauf ausgeschlossen.

Profilname

Gibt das Inventory-Profil an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Hinweise

Mit den Optionen **–t** und **–d** können Sie je eine getrennte Liste mit Verzeichnissen, die ausgeschlossen werden sollen, und mit Verzeichnissen, die eingeschlossen werden sollen, erstellen. Dies ist für die Liste der Dateien oder Dateitypen (Option **–f**) jedoch nicht möglich. Für Dateien und Dateitypen können Sie mit der Option **–e** angeben, ob die gesamte Liste beim Suchlauf eingeschlossen oder ausgeschlossen werden soll.

Beispiele

Im folgenden Beispiel wird das Verzeichnis ’/cache’ der Liste der Verzeichnisse, die beim Softwaresuchlauf ausgeschlossen werden sollen, hinzugefügt. Diese Änderung wird auf das Inventory-Profil ’SW_Profile’ angewendet.

```
wsetinvpcfiles -t EXCLUDE -d +/cache @InventoryConfig:SW_Profile
```

Im folgenden Beispiel werden die Verzeichnisse ’/tmp’ und ’/temp’ aus der Liste der Verzeichnisse, die beim Softwaresuchlauf ausge-

schlossen werden sollen, entfernt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvpcfiles -t EXCLUDE -d -"*/tmp" -d-"*/temp" \
@InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird das Verzeichnis '/usr' aus der Liste der Verzeichnisse, die beim Softwaresuchlauf eingeschlossen werden sollen, entfernt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvpcfiles -t INCLUDE -d -/usr @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird die Liste der Dateien und Dateitypen, die mit der Option **–f** angegeben wird, beim Suchlauf ausgeschlossen. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvpcfiles -e EXCLUDE @InventoryConfig:SW_Profile
```

Im folgenden Beispiel werden die Dateitypen '.bat' und '.sys' den Dateien, die beim Suchlauf gefiltert werden sollen, hinzugefügt. Ob diese Dateien beim Suchlauf eingeschlossen oder ausgeschlossen werden, hängt von der Einstellung für die Option **–e** ab. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvpcfiles -f +"*.bat" –f +"*.sys" @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird die Datei 'TEST.BAT' bei der Profilverteilung ausgeführt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

Anmerkung: Da der Pfad für die Datei 'TEST.BAT' in diesem Beispiel nicht angegeben wird, muss sie sich in dem Verzeichnis befinden, in dem der Befehl ausgeführt wird.

```
wsetinvpcfiles -s "TEST.BAT" @InventoryConfig:SW_Profile
```

Weitere Informationen

wgetinvpcfiles, **wsetinvglobal**, **wsetinvpchw**, **wsetinvpcsw**, **wsetinvunixfiles**, **wsetinvunixhw**, **wsetinvunixsw**

wsetinvpchw

Mit diesem Befehl werden die Optionen für Hardwaresuchläufe auf PC-Endpunkten festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvpchw –a *Komponente Profilname*

wsetinvpchw –r *Komponente Profilname*

wsetinvpchw –t {**Y** | **N**} *Profilname*

wsetinvpchw –u {**Y** | **N**} *Profilname*

Beschreibung

Mit dem Befehl **wsetinvpchw** werden die Optionen für einen Hardwaresuchlauf auf einem PC-Endpunkt festgelegt. Sie können folgende Optionen angeben:

- Ob bei Suchläufen auf PC-Endpunkten die Tivoli Hardware-Suchfunktion ausgeführt werden soll.
- Ob die Suchlaufdaten an das Konfigurations-Repository gesendet oder für Datensammlungen zu einem späteren Zeitpunkt auf dem Endpunkt gespeichert werden sollen.
- Nach welchen Hardwarekomponenten gesucht werden soll.

Optionen

–a *Komponente*

Fügt die angegebene Komponente der Liste mit Hardwarekomponenten hinzu, nach denen gesucht werden soll. Sie können folgende Komponenten angeben:

Processor

Beschreibt den oder die auf dem System installierten Prozessoren.

OperatingSystem
Beschreibt nur das Betriebssystem, das zum Zeitpunkt des Suchlaufs aktiv ist. Mit dieser Option werden nicht alle auf einem Dual-boot-System installierten Betriebssysteme aufgelistet.

IPAddress
Listet die TCP/IP-Konfigurationsdaten auf.

Partition
Beschreibt die einzelnen Laufwerkspartitionen des Systems. Es kann auch Systeme mit nur einer Partition geben. Zu den aufgeführten allgemeinen Informationen können die Laufwerkszuordnungen, die Größe sowie der verfügbare Speicherplatz der jeweiligen Partitionen, das Dateisystem und der Datenträgertyp gehören.

Keyboard
Beschreibt die angeschlossene Tastatur (falls vorhanden).

IPXAddress
Listet die IPX-Konfigurationsdaten auf.

Printer
Beschreibt lokale Drucker und Netzwerkdrucker, die am System angeschlossen sind.

Memory
Beschreibt den vorhandenen Arbeitsspeicher. Mit dieser Option können Angaben zur tatsächlichen Arbeitsspeichergröße, zum virtuellen Arbeitsspeicher und zum Auslagerungsbereich aufgelistet werden.

Storage
Beschreibt alle lokalen Speichergeräte wie z. B. Festplatten, CD-ROM-Laufwerke und

Laufwerke für austauschbare Datenträger. Mit dieser Option werden keine Netzlaufwerke aufgelistet.

NetworkAdapter
Beschreibt den oder die installierten Netzadapter.

PointingDev
Beschreibt die an das System angeschlossene Zeigereinheit (wie z. B. Maus oder Rollkugel).

Video Beschreibt den Videoadapter und die aktuellen Videoeinstellungen. Mit dieser Option werden nicht alle Einstellungen angezeigt.

USBDev
Beschreibt alle zum Zeitpunkt des Suchlaufs an das System angeschlossenen USB-Geräte und Hubs.

PCI Listet die Erweiterungskarten auf, die am PCI- und AGP-Bus installiert sind.

PCSystemParams
Listet PC-spezifische Informationen wie Benutzername, Domänenname und BIOS-Eigenschaften auf.

Anmerkung: Bei diesen Komponentenwerten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie brauchen auch nicht den gesamten Komponentennamen einzugeben. Sie brauchen die Namen nur bis zu der Anzahl von Zeichen einzugeben, mit der die jeweilige Komponente eindeutig ermittelt werden kann. Sie können die Option **–a** *Komponente* mehrmals verwenden, um mehrere Komponenten anzugeben.

–r *Komponente*

Entfernt die angegebene Komponente aus der Liste mit Hardwarekomponenten, nach denen gesucht werden soll. Für *Komponente* können dieselben Werte wie bei der Option **–a** verwendet werden. Sie können die Option **–r** *Komponente* mehrmals verwenden, um mehrere Komponenten anzugeben.

–t *Option*

Gibt an, ob die Tivoli Hardware-Suchfunktion ausgeführt werden soll. Sie können eine der folgenden Optionen verwenden:

Y Die Tivoli Hardware-Suchfunktion wird ausgeführt.

N Die Tivoli Hardware-Suchfunktion wird nicht ausgeführt.

–u *Option*

Gibt an, ob das Konfigurations-Repository anhand der Suchlaufdaten aktualisiert werden soll. Sie können eine der folgenden Optionen verwenden:

Y Der Suchlauf wird ausgeführt, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

N Die Suchlaufdaten werden nicht an das Konfigurations-Repository gesendet. Der Suchlauf wird ausgeführt, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

Profilname

Gibt das Profil an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Mit folgendem Befehl wird das Inventory-Profil 'HW_Profile' so konfiguriert, dass die Tivoli Hardware-Suchfunktion ausgeführt wird, Informationen zu Prozessoren beim Suchlauf ausgeschlossen und Informationen zum Hauptspeicher und zu Laufwerkspartitionen eingeschlossen werden.

```
wsetinvpchw -r Processor -a Memory -a Partition -t Y \
@InventoryConfig:HW_Profile
```

Weitere Informationen

wgetinvpchw, **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpcsw**, **wsetinvunixfiles**, **wsetinvunixhw**, **wsetinvunixsw**

wsetinvpcsw

Mit diesem Befehl werden die Optionen für Softwaresuchläufe auf PC-Endpunkten festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvpcsw –b {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvpcsw –c {**QUICK** | **FULL** | **MD5** | **NONE**} *Profilname*

wsetinvpcsw –f {**Y** | **N**} *Profilname*

wsetinvpcsw –h {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvpcsw –r {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvpcsw –s {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvpcsw –x {**Y** | **N**} *Profilname*

Beschreibung

Mit dem Befehl **wsetinvpcsw** werden die Optionen für Softwaresuchläufe auf PC-Endpunkten festgelegt. Sie können angeben, ob die Dateien auf einem PC-Endpunkt durchsucht werden sollen, ob die Windows-Registrierdatenbank nach Informationen zu installierten Programmen durchsucht werden soll oder ob beide Vorgänge ausgeführt werden sollen.

Bei einem Suchlauf in Dateien können Sie festlegen, ob dabei folgende Aktionen ausgeführt werden sollen:

- Nach Softwaresignaturdaten suchen.
- Nach grundlegenden Dateiinformationen suchen.
- Angepassten Filter für Suchläufe nach grundlegenden Dateiinformationen verwenden.
- Dateien nach Header-Informationen durchsuchen.

- Für die durchsuchten Dateien Kontrollsummen generieren.
- Suchlaufdaten an das Konfigurations-Repository senden.
- Nur ausführbare Dateien durchsuchen.

Bei Suchläufen in der Windows-Registrierdatenbank können Sie angeben, ob die Suchlaufdaten an das Konfigurations-Repository gesendet werden sollen.

Optionen

–b *Option* Gibt an, ob nach grundlegenden Dateiinformationen gesucht werden soll. Bei dieser Art Suchlauf werden Name und Pfad aller durchsuchten Dateien gesammelt. Auf manchen Plattformen werden noch weitere Informationen erfasst, wie z. B. Erstellungsdatum, Änderungsdatum und Datum des letzten Zugriffs. Sie können eine der folgenden Optionen verwenden:

SCAN Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs nach grundlegenden Dateiinformationen an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach grundlegenden Dateiinformationen ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–c *Option* Gibt an, ob Kontrollsummen für die durchsuchten Dateien generiert werden sollen. Sie können eine der folgenden Optionen verwenden:

QUICK
Generiert den schnellen Kontrollsummenwert. Mit dieser Option wird ein 32-Bit-Wert erstellt, der nur auf den ersten 1 KB jeder Datei basiert. Dies ist die schnellste Option für Kontrollsummen, sie erzeugt aber den am wenigsten zuverlässigen Wert.

FULL Generiert den vollständigen Kontrollsummenwert. Mit dieser Option wird ein 32-Bit-Wert erstellt, der auf dem gesamten Inhalt jeder Datei basiert. Diese Option ist langwieriger als die Option für den schnellen Kontrollsummenwert, sie ergibt aber einen zuverlässigeren Wert.

MD5 Generiert den MD5-Kontrollsummenwert. Mit dieser Option wird ein 128-Bit-Wert erstellt, der auf dem gesamten Inhalt jeder Datei basiert. Der von der Option 'MD5' gelieferte Wert ist zuverlässiger als der Wert der schnellen oder der vollständigen Kontrollsumme, seine Erzeugung benötigt jedoch mehr Zeit.

NONE
Es werden keine Kontrollsummenwerte generiert.

–f *Option* Gibt an, ob die Suche nach grundlegenden Dateiinformationen mit einer angepassten Dateiliste gefiltert, also eingeschränkt werden soll. Sie können diese Dateiliste entweder im Fenster 'Filter anpassen' oder mit dem Befehl **winvfilter** erstellen. Wenn Sie für diese Option den Wert 'Y' angeben, werden bei dem Suchlauf nach grundlegenden Dateiinformationen nur

die im angepassten Filter angegebenen Dateien eingeschlossen. Sie können eine der folgenden Optionen verwenden:

Y Suchläufe nach grundlegenden Dateiinformationen werden anhand des angepassten Filters eingeschränkt.

N Suchläufe nach grundlegenden Dateiinformationen werden nicht anhand des angepassten Filters eingeschränkt.

–h *Option* Gibt an, ob im Header von durchsuchten Dateien nach dem Firmennamen, Produktnamen oder der Produktversion gesucht werden soll. Diese Option bleibt auf NetWare- und OS/2-Systemen ohne Wirkung. Sie können eine der folgenden Optionen verwenden:

SCAN Der Endpunkt wird nach Header-Daten durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs nach Header-Informationen an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach Header-Informationen durchsucht, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach Header-Informationen ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–r *Option* Gibt an, ob die Windows-Registrierdatenbank nach Informationen zu installierten Produkten durchsucht werden soll. Diese Option bleibt auf NetWare- und

OS/2-Systemen ohne Wirkung. Sie können eine der folgenden Optionen verwenden:

SCAN Die Windows-Registrierdatenbank wird durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs in der Windows-Registrierdatenbank an das Konfigurations-Repository gesendet.

BOTH
Die Windows-Registrierdatenbank wird durchsucht, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf in der Windows-Registrierdatenbank ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–s Gibt an, ob mit Hilfe von Softwaresignaturen gesucht werden soll. Bei dieser Suchlaufart werden Name und Größe jeder durchsuchten Datei mit der Liste der Softwaresignaturen verglichen. Sie können eine der folgenden Optionen verwenden:

SCAN Der Suchlauf wird mit Hilfe von Softwaresignaturen ausgeführt, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs mit Hilfe von Softwaresignaturen an das Konfigurations-Repository gesendet.

BOTH
Der Suchlauf wird mit Hilfe von Software-

signaturen ausgeführt, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf mit Hilfe von Softwaresignaturen ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–x *Option* Gibt an, ob der Suchlauf in Dateien eingeschränkt werden soll, indem nur ausführbare Dateien durchsucht werden. Bei PC-Systemen sind Dateien mit einer der folgenden Erweiterung als ausführbare Dateien definiert: 'EXE', 'COM', 'CMD', 'BAT'. Sie können eine der folgenden Optionen verwenden:

Y Es werden nur ausführbare Dateien durchsucht.

N Der Suchlauf wird nicht auf ausführbare Dateien beschränkt.

Profilname Gibt das Inventory-Profil an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Mit folgendem Befehl wird das Inventory-Profil 'SW_Profile' so konfiguriert, dass eine MD5-Kontrollsumme der durchsuchten Dateien generiert, die Windows-Registrierdatenbank durchsucht und die Suchlaufdaten aus der Registrierdatenbank an das Konfigurations-Repository gesendet werden.

```
wsetinvpcsw -c MD5 -r BOTH @InventoryConfig:SW_Profile
```

Weitere Informationen

wgetinvpcsw, **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvunixfiles**, **wsetinvunixhw**, **wsetinvunixsw**

wsetinvunixfiles

Mit diesem Befehl werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, angegeben.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvunixfiles –e {**INCLUDE** | **EXCLUDE**} *Profilname*

wsetinvunixfiles –f {+ | –}*Dateityp Profilname*

wsetinvunixfiles –m {+ | –}*MIF-Pfad Profilname*

wsetinvunixfiles –s *Skriptpfad Profilname*

wsetinvunixfiles –t {**INCLUDE** | **EXCLUDE**} **–d** {+ | –}*Verzeichnis Profilname*

Beschreibung

Mit dem Befehl **wsetinvunixfiles** werden die Optionen für zu durchsuchende Dateien und Verzeichnisse, auszuführende Skripts und benutzerdefinierte MIF-Dateien, die bei Suchläufen auf UNIX-Endpunkten gesammelt werden sollen, angegeben. Mit diesem Befehl können Sie folgende Optionen festlegen:

- Die Dateien oder Dateitypen, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
- Die Verzeichnisse, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen.
- Die Skripts, die nach Abschluss des Suchlaufs bei der Profilverteilung ausgeführt werden sollen.
- Die benutzerdefinierten MIF-Dateien, die nach Abschluss des Suchlaufs bei der Profilverteilung gesammelt werden sollen.

Optionen

–d *Option Verzeichnis*

Gibt ein Verzeichnis an, das in der Liste der Verzeichnisse, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen, hinzugefügt oder entfernt werden soll. Standardmäßig wird das Verzeichnis '*/tmp' beim Suchlauf ausgeschlossen. Um anzuzeigen, ob das Verzeichnis beim Suchlauf eingeschlossen oder ausgeschlossen werden soll, müssen Sie dieser Option die Option **–t** voranstellen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Das Verzeichnis wird der Liste hinzugefügt.

– Das Verzeichnis wird aus der Liste entfernt.

Bei Angabe eines Verzeichnisses, das in einen Suchlauf *eingeschlossen* werden soll, müssen die folgenden Hinweise befolgt werden:

- Sie können keine Platzhalterzeichen verwenden. Alle Zeichen werden als Literale behandelt, auch die Zeichen Stern (*) und Fragezeichen (?).
- Alle Verzeichnisse werden als absolute Pfadangaben behandelt. Bei der Angabe von 'abc' wird '/abc' durchsucht, nicht aber '/tmp/abc'.

Bei Angabe eines Verzeichnisses, das von einem Suchlauf *ausgeschlossen* werden soll, müssen die folgenden Hinweise beachtet werden:

- Sie können am Anfang oder Ende eines Pfadnamens einen Stern als Platzhalterzeichen verwenden, z. B. '*abc', '*/abc' oder 'abc/def*'.
- Innerhalb eines Pfadnamens können Sie kein Platzhalterzeichen verwenden, z. B. /*/tmp, t*mp, oder /usr*/tmp). Alle Zeichen innerhalb von Pfadangaben werden als Literale behandelt.

- Verzeichnisse, die einen Stern (*) enthalten, müssen in Anführungszeichen (″ ″) gesetzt werden.

–e *Option*

Gibt an, ob die Liste der Dateien oder Dateitypen, die mit der Option **–f** erstellt wurde, beim Suchlauf eingeschlossen oder ausgeschlossen werden soll. Sie können eine der folgenden Optionen verwenden:

INCLUDE

Die Dateien oder Dateitypen werden beim Suchlauf eingeschlossen. Dies ist die Standardoption.

EXCLUDE

Die Dateien oder Dateitypen werden beim Suchlauf ausgeschlossen.

–f *Option Dateityp*

Gibt eine Datei oder einen Dateityp an, die oder der in der Liste der Dateien oder Dateitypen, die beim Suchlauf eingeschlossen oder ausgeschlossen werden sollen, hinzugefügt oder entfernt werden soll. Sie können eine Datei mit Hilfe des Dateinamens bzw. einen Dateityp mit Hilfe eines Sterns (*) als Platzhalterzeichen angeben. Sie können den Stern an den Anfang, an das Ende oder an beide Positionen der Zeichenfolge stellen (Beispiele: ’*.txt’, ’abc.*’, ’*abc.txt’, ’abc.tx*’, ’*abc’, ’abc*’, ’*abc*’). Innerhalb der Zeichenfolge können Sie jedoch keinen Stern verwenden (Beispiele: ’ab*c’, ’ab*c.txt’, ’abc.t*t’). Dateitypen müssen in Anführungszeichen (″ ″) gesetzt werden. Sie können die Option **–f** mehrmals verwenden, um mehrere Dateitypen hinzuzufügen oder zu entfernen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Die Datei oder der Dateityp wird der Liste hinzugefügt.

– Die Datei oder der Dateityp wird aus der Liste entfernt.

–m *Option MIF-Pfad*

Gibt die Position und den Namen einer MIF-Datei an, die in der Liste der MIF-Dateien, die bei der Profilverteilung gesammelt werden sollen, hinzugefügt oder entfernt werden soll. Für die Option *MIF-Pfad* muss der vollständige Pfad und Name der MIF-Datei angegeben werden. Sie können die Option **–m** mehrmals verwenden, um mehrere MIF-Dateien hinzuzufügen oder zu entfernen. Sie können eine der folgenden Optionen verwenden. Zwischen der Option und dem zugeordneten Wert darf kein Leerzeichen eingegeben werden.

+ Die MIF-Datei wird der Liste hinzugefügt.

– Die MIF-Datei wird aus der Liste entfernt.

–s *Skriptpfad*

Gibt die Position und den Namen eines Skripts an, das bei der Profilverteilung auf einem Endpunkt ausgeführt werden soll. Das Skript wird nach Abschluss des Suchlaufs, jedoch vor dem Lesen der MIF-Dateien ausgeführt. Es können nur Textdateien angegeben werden. Die Angabe von Binärdateien ist nicht zulässig.

Anmerkungen:

1. Wenn Sie ein Skript editieren, nachdem Sie es hinzugefügt haben, muss es anschließend erneut mit der Option **–s** hinzugefügt werden. Wenn Sie diesen Schritt nicht ausführen, wird das editierte Skript nicht ausgeführt.
2. Wenn ein Skript entfernt werden soll, geben Sie Anführungszeichen ohne Zeichen dazwischen (″″) ein.

–t *Option*

Gibt an, ob das Verzeichnis, das mit der im Anschluss an diese Option eingegebenen Option **–d** angegeben wird, bei Softwaresuchläufen auf UNIX-Systemen eingeschlossen oder ausgeschlossen werden soll. Sie können eine der folgenden Optionen verwenden:

INCLUDE

Das angegebene Verzeichnis wird beim Suchlauf eingeschlossen.

EXCLUDE

Das angegebene Verzeichnis wird beim Suchlauf ausgeschlossen.

Profilname

Gibt den Namen des Inventory-Profils an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Hinweise

Mit den Optionen **–t** und **–d** können Sie je eine getrennte Liste mit Verzeichnissen, die ausgeschlossen werden sollen, und mit Verzeichnissen, die eingeschlossen werden sollen, erstellen. Dies ist für die Liste der Dateien oder Dateitypen (Option **–f**) jedoch nicht möglich. Für Dateien und Dateitypen können Sie mit der Option **–e** angeben, ob die gesamte Liste beim Suchlauf eingeschlossen oder ausgeschlossen werden soll.

Beispiele

Im folgenden Beispiel wird das Verzeichnis '/cache' der Liste der Verzeichnisse, die beim Softwaresuchlauf ausgeschlossen werden sollen, hinzugefügt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvunixfiles -t EXCLUDE -d +/cache @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird das Verzeichnis '/tmp' aus der Liste der Verzeichnisse, die beim Softwaresuchlauf ausgeschlossen werden sollen, entfernt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvunixfiles -t EXCLUDE -d -"*/tmp" \
@InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird das Verzeichnis '/usr' aus der Liste der Verzeichnisse, die beim Softwaresuchlauf eingeschlossen werden sollen, entfernt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvunixfiles -t INCLUDE -d -/usr @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird die Liste der Dateien und Dateitypen, die mit der Option **–f** angegeben wird, beim Suchlauf ausgeschlossen. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvunixfiles -e EXCLUDE @InventoryConfig:SW_Profile
```

Im folgenden Beispiel werden die Dateitypen '.txt' und '.gif' den Dateien, die beim Suchlauf gefiltert werden sollen, hinzugefügt. Ob diese Dateien beim Suchlauf eingeschlossen oder ausgeschlossen werden, hängt von der Einstellung für die Option **–e** ab. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

```
wsetinvunixfiles -f +"*.txt" +"*.gif" @InventoryConfig:SW_Profile
```

Im folgenden Beispiel wird das Skript 'create_mif.sh' bei der Profilverteilung ausgeführt. Diese Änderung wird auf das Inventory-Profil 'SW_Profile' angewendet.

Anmerkung: Da der Pfad für das Skript 'create_mif.sh' in diesem Beispiel nicht angegeben wird, muss es sich in dem Verzeichnis befinden, in dem der Befehl ausgeführt wird.

```
wsetinvunixfiles -s "create_mif.sh" @InventoryConfig:SW_Profile
```

Weitere Informationen

wgetinvunixfiles, **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvpcsw**, **wsetinvunixhw**, **wsetinvunixsw**

wsetinvunixhw

Mit diesem Befehl werden die Optionen für Hardwaresuchläufe auf UNIX-Endpunkten festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvunixhw –a *Komponente Profilname*

wsetinvunixhw –r *Komponente Profilname*

wsetinvunixhw –t {**Y** | **N**} *Profilname*

wsetinvunixhw –u {**Y** | **N**} *Profilname*

Beschreibung

Mit dem Befehl **wsetinvunixhw** werden die Optionen für einen Hardwaresuchlauf auf einem UNIX-Endpunkt festgelegt. Sie können folgende Optionen angeben:

- Ob bei Suchläufen auf UNIX-Endpunkten die Tivoli Hardware-Suchfunktion ausgeführt werden soll.
- Ob die Suchlaufdaten an das Konfigurations-Repository gesendet oder für Datensammlungen zu einem späteren Zeitpunkt auf dem Endpunkt gespeichert werden sollen.
- Nach welchen Hardwarekomponenten gesucht werden soll.

Optionen

–a *Komponente*

Fügt die angegebene Komponente der Liste mit Hardwarekomponenten hinzu, nach denen gesucht werden soll. Sie können folgende Komponenten angeben:

Processor

Beschreibt den oder die auf dem System installierten Prozessoren.

OperatingSystem
Beschreibt nur das Betriebssystem, das zum Zeitpunkt des Suchlaufs aktiv ist. Mit dieser Option werden nicht alle auf einem Dual-boot-System installierten Betriebssysteme aufgelistet.

IPAddress
Listet die TCP/IP-Konfigurationsdaten auf.

Partition
Beschreibt die einzelnen Laufwerkspartitionen des Systems. Es kann auch Systeme mit nur einer Partition geben. Zu den aufgeführten allgemeinen Informationen können die Laufwerkszuordnungen, die Größe sowie der verfügbare Speicherplatz der jeweiligen Partitionen, das Dateisystem und der Datenträgertyp gehören.

Keyboard
Beschreibt die angeschlossene Tastatur (falls vorhanden).

Memory
Beschreibt den vorhandenen Arbeitsspeicher. Mit dieser Option können Angaben zur tatsächlichen Arbeitsspeichergröße, zum virtuellen Arbeitsspeicher und zum Auslagerungsbereich aufgelistet werden.

Storage
Beschreibt alle lokalen Speichergeräte wie z. B. Festplatten, CD-ROM-Laufwerke und Laufwerke für austauschbare Datenträger. Mit dieser Option werden keine Netzlaufwerke aufgelistet.

NetworkAdapter
Beschreibt den oder die installierten Netzadapter.

PointingDev
Beschreibt die an das System angeschlossene Zeigereinheit (wie z. B. Maus oder Rollkugel).

UnixSystemParam
Beschreibt den aktuellen Status des Systems, z. B. die Boot-Zeit und die Ausführungsebene.

Anmerkung: Bei diesen Komponentenwerten muss die Groß-/Kleinschreibung nicht beachtet werden. Sie brauchen auch nicht den gesamten Komponentennamen einzugeben. Sie brauchen die Namen nur bis zu der Anzahl von Zeichen einzugeben, mit der die jeweilige Komponente eindeutig ermittelt werden kann. Sie können die Option **–a** *Komponente* mehrmals verwenden, um mehrere Komponenten anzugeben.

–r *Komponente*
Entfernt die angegebene Komponente aus der Liste mit Hardwarekomponenten, nach denen gesucht werden soll. Für *Komponente* können dieselben Werte wie bei der Option **–a** verwendet werden. Sie können die Option **–r** *Komponente* mehrmals verwenden, um mehrere Komponenten anzugeben.

–t *Option* Gibt an, ob die Tivoli Hardware-Suchfunktion ausgeführt werden soll. Sie können eine der folgenden Optionen verwenden:

Y Die Tivoli Hardware-Suchfunktion wird ausgeführt.

N Die Tivoli Hardware-Suchfunktion wird nicht ausgeführt.

–u *Option* Gibt an, ob das Konfigurations-Repository anhand der Suchlaufdaten aktualisiert werden soll. Sie können eine der folgenden Optionen verwenden:

Y Der Suchlauf wird ausgeführt, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

N Die Suchlaufdaten werden nicht an das Konfigurations-Repository gesendet. Der Suchlauf wird ausgeführt, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

Profilname Gibt den Namen des Inventory-Profils an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Mit folgendem Befehl wird das Inventory-Profil 'HW_Profile' so konfiguriert, dass die Tivoli Hardware-Suchfunktion ausgeführt wird, UNIX-Systemparameter beim Suchlauf eingeschlossen und die Suchlaufdaten auf dem Endpunkt gespeichert werden.

```
wsetinvunixhw -a UNIXSystemParam -t Y -u N \
@InventoryConfig:HW_Profile
```

Weitere Informationen

wgetinvunixhw, **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvpcsw**, **wsetinvunixfiles**, **wsetinvunixsw**

wsetinvunixsw

Mit diesem Befehl werden die Optionen für Softwaresuchläufe auf UNIX-Endpunkten festgelegt.

Syntax

Jede Befehlsoption wird in einer eigenen Zeile angegeben, es sei denn, die Optionen müssen zusammen verwendet werden.

wsetinvunixsw –b {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvunixsw –c {**QUICK** | **FULL** | **MD5** | **NONE**} *Profilname*

wsetinvunixsw –f {**Y** | **N**} *Profilname*

wsetinvunixsw –p {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvunixsw –s {**SCAN | UPDATE | BOTH** | **NO**} *Profilname*

wsetinvunixsw –x {**Y** | **N**} *Profilname*

Beschreibung

Mit dem Befehl **wsetinvunixsw** werden die Optionen für Softwaresuchläufe auf UNIX-Endpunkten festgelegt. Sie können angeben, ob die Dateien auf einem UNIX-Endpunkt durchsucht werden sollen, ob das Betriebssystem nach Informationen zu installierten Programmen durchsucht werden soll oder ob beide Vorgänge ausgeführt werden sollen.

Bei einem Suchlauf in Dateien können Sie festlegen, ob dabei folgende Aktionen ausgeführt werden sollen:

- Nach Softwaresignaturdaten suchen.
- Nach grundlegenden Dateiinformationen suchen.
- Angepassten Filter für Suchläufe nach grundlegenden Dateiinformationen verwenden.
- Für die durchsuchten Dateien Kontrollsummen generieren.
- Suchlaufdaten an das Konfigurations-Repository senden.

- Nur ausführbare Dateien durchsuchen.

Bei Suchläufen im Betriebssystem können Sie angeben, ob die Suchlaufdaten an das Konfigurations-Repository gesendet werden sollen.

Optionen

–b *Option* Gibt an, ob nach grundlegenden Dateiinformationen gesucht werden soll. Bei dieser Art Suchlauf werden Name und Pfad aller durchsuchten Dateien gesammelt. Auf manchen Plattformen werden noch weitere Informationen erfasst, wie z. B. Erstellungsdatum, Änderungsdatum und Datum des letzten Zugriffs. Sie können eine der folgenden Optionen verwenden:

SCAN Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs nach grundlegenden Dateiinformationen an das Konfigurations-Repository gesendet.

BOTH
Der Endpunkt wird nach grundlegenden Dateiinformationen durchsucht, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf nach grundlegenden Dateiinformationen ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–c *Option* Gibt an, ob Kontrollsummen für die durchsuchten Dateien generiert werden sollen. Sie können eine der folgenden Optionen verwenden:

QUICK
Generiert den schnellen Kontrollsummen-

wert. Mit dieser Option wird ein 32-Bit-Wert erstellt, der nur auf den ersten 1 KB jeder Datei basiert. Dies ist die schnellste Option für Kontrollsummen, sie erzeugt aber den am wenigsten zuverlässigen Wert.

FULL Generiert den vollständigen Kontrollsummenwert. Mit dieser Option wird ein 32-Bit-Wert erstellt, der auf dem gesamten Inhalt jeder Datei basiert. Diese Option ist langwieriger als die Option für den schnellen Kontrollsummenwert, sie ergibt aber einen zuverlässigeren Wert.

MD5 Generiert den MD5-Kontrollsummenwert. Mit dieser Option wird ein 128-Bit-Wert erstellt, der auf dem gesamten Inhalt jeder Datei basiert. Der von der Option 'MD5' gelieferte Wert ist zuverlässiger als der Wert der schnellen oder der vollständigen Kontrollsumme, seine Erzeugung benötigt jedoch mehr Zeit.

NONE

Es werden keine Kontrollsummenwerte generiert.

–f *Option* Gibt an, ob die Suche nach grundlegenden Dateiinformationen mit einer angepassten Dateiliste gefiltert, also eingeschränkt werden soll. Sie können diese Dateiliste entweder im Fenster 'Filter anpassen' oder mit dem Befehl **winvfilter** erstellen. Wenn Sie für diese Option den Wert 'Y' angeben, werden bei dem Suchlauf nach grundlegenden Dateiinformationen nur die im angepassten Filter angegebenen Dateien eingeschlossen. Sie können eine der folgenden Optionen verwenden:

Y Suchläufe nach grundlegenden Dateiinformationen werden anhand des angepassten Filters eingeschränkt.

N Suchläufe nach grundlegenden Dateiinformationen werden nicht anhand des angepassten Filters eingeschränkt.

–p *Option* Gibt an, ob das Betriebssystem nach Informationen zu installierten Produkten und Programmkorrekturen durchsucht werden soll. Sie können eine der folgenden Optionen verwenden:

SCAN Das Betriebssystem wird durchsucht, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs im Betriebssystem an das Konfigurations-Repository gesendet.

BOTH
Das Betriebssystem wird durchsucht, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf im Betriebssystem ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–s Gibt an, ob mit Hilfe von Softwaresignaturen gesucht werden soll. Bei dieser Suchlaufart werden Name und Größe jeder durchsuchten Datei mit der Liste der Softwaresignaturen verglichen. Sie können eine der folgenden Optionen verwenden:

SCAN Der Suchlauf wird mit Hilfe von Softwaresignaturen ausgeführt, und die Suchlaufdaten werden anschließend auf dem Endpunkt gespeichert.

UPDATE
Es wird kein Suchlauf ausgeführt. Stattdessen werden die Daten des letzten Suchlaufs mit Hilfe von Softwaresignaturen an das Konfigurations-Repository gesendet.

BOTH
Der Suchlauf wird mit Hilfe von Softwaresignaturen ausgeführt, und die Suchlaufdaten werden anschließend an das Konfigurations-Repository gesendet.

NO Es wird kein Suchlauf mit Hilfe von Softwaresignaturen ausgeführt, und es werden auch keine Daten an das Konfigurations-Repository gesendet.

–x *Option* Gibt an, ob der Suchlauf in Dateien eingeschränkt werden soll, indem nur ausführbare Dateien durchsucht werden. Bei UNIX-Systemen sind solche Dateien als ausführbare Dateien definiert, bei denen die Ausführungsberechtigung gesetzt ist. Sie können eine der folgenden Optionen verwenden:

Y Es werden nur ausführbare Dateien durchsucht.

N Der Suchlauf wird nicht auf ausführbare Dateien beschränkt.

Profilname Gibt den Namen des Inventory-Profils an, das Sie editieren möchten. Geben Sie den Profilnamen in folgendem Format an: **@InventoryConfig:***Profilname*.

Autorisierung

admin, **senior**, **super** oder **Inventory_edit**

Beispiele

Mit folgendem Befehl wird das Inventory-Profil 'SW_Profile' so konfiguriert, dass eine schnelle Kontrollsumme der durchsuchten Dateien generiert wird, das Betriebssystem nach Informationen zu installierten Produkten und Programmkorrekturen durchsucht wird und die Suchlaufdaten aus dem Betriebssystem an das Konfigurations-Repository gesendet werden.

```
wsetinvunixsw -c QUICK -p BOTH @InventoryConfig:SW_Profile
```

Weitere Informationen

wgetinvunixsw, **wsetinvglobal**, **wsetinvpcfiles**, **wsetinvpchw**, **wsetinvunixfiles**, **wsetinvunixhw**, **wsetinvpcsw**

C

Konfigurations-Repository

In diesem Anhang sind alle Ansichten und Tabellen aufgeführt und erläutert, die im Schema des Konfigurations-Repositorys von Tivoli Inventory Version 4.0 enthalten sind. Darüber hinaus sind hier alle Abfragen aufgeführt und erläutert, die bei der Installation von Inventory Version 4.0 installiert werden können. Bei der Installation von Tivoli Inventory wird das Konfigurations-Repository von Tivoli Inventory auf dem RDBMS-Server erstellt. In diesem Konfigurations-Repository werden die bei Inventory-Suchläufen erfassten Daten gespeichert.

Anmerkung: Die Angaben im vorliegenden Anhang zur Definition der Tabellen und Ansichten im Konfigurations-Repository von Tivoli Inventory können Sie auch den Systemkatalogen Ihrer RDBMS-Datenbank entnehmen. Weitere Informationen hierzu können Sie den RDBMS-Handbüchern entnehmen, oder wenden Sie sich an Ihren Datenbankadministrator.

Eine grafische Darstellung der Tabellen im Konfigurations-Repository von Tivoli Inventory finden Sie auf dem Poster *Tivoli Inventory Configuration Repository*, das mit dem vorliegenden Handbuch geliefert wird; ggf. können Sie eine Kopie über den Tivoli-Kundendienst erhalten. Sie haben auch die Möglichkeit, eigene Tabellen und Ansichten zu erstellen.

Anweisungen zum Hinzufügen von Tabellen im Konfigurations-Repository können Sie dem Kapitel „Benutzerspezifische Daten mit

Tivoli Inventory erfassen" auf Seite 121 entnehmen. Anweisungen zur Erstellung von Abfragen finden Sie im Kapitel „Inventory-Daten abfragen" auf Seite 209. Weitere Informationen zum Löschen von Daten aus dem Konfigurations-Repository unter Verwendung des Befehls **winvrmnode** können Sie dem Anhang „Befehle" auf Seite 239 entnehmen.

Protokollfunktion

In Tivoli Inventory Version 3.6.*x* wurden nur die Daten aus dem aktuellen und dem letzten Suchlauf gespeichert. Tivoli Inventory Version 4.0 verfügt jedoch über eine optionale Protokollfunktion, mit der Daten aus allen vorherigen Suchläufen gespeichert und geändert werden können. Die Daten aus dem aktuellen Suchlauf werden in Betriebsdatentabellen gespeichert. Betriebsdatentabellen werden bei jedem Suchlauf überschrieben oder aktualisiert, je nachdem, ob für das Inventory-Profil die Option 'Nur anhand der Änderungen aktualisieren' oder 'Durch aktuelle Ergebnisse ersetzen' ausgewählt ist. Betriebsdatentabellen spiegeln nur den aktuellsten Suchlauf wider. Wenn Sie jedoch die Protokollfunktion aktivieren, werden neue, geänderte und gelöschte Daten aus den Betriebsdatentabellen in Protokolltabellen gespeichert, während die Betriebsdatentabellen überschrieben werden.

Sie können auf die Protokolldaten über Protokollansichten, -abfragen und -tabellen zugreifen. Die Namen der Protokollansichten, -abfragen und -tabellen beginnen mit einem großen H und einem Unterstrich (H_). Darauf folgt der Name der Betriebsansicht, -abfrage oder -tabelle, der die entsprechende Protokollansicht, -abfrage oder -tabelle zugeordnet ist.

Jede Betriebsdatentabelle, die die Spalte COMPUTER_SYS_ID enthält, verfügt über eine entsprechende Protokolltabelle. Eine Protokolltabelle enthält alle Spaltennamen der entsprechenden Betriebsdatentabelle sowie zwei zusätzliche Spalten, RECORD_ACTION und PRFL_ACTION. Die Spalte RECORD_ACTION kann folgende Werte enthalten: INSERT (neue Informationen werden zur Betriebsdatentabelle hinzugefügt), UPDATE (ein Datensatz in der Betriebsdatentabelle wird teilweise geändert) oder DELETE (der

Datensatz ist in der Betriebsdatentabelle nicht mehr vorhanden). PRFL_ACTION gibt an, ob als Profilkonfigurationsoption 'Durch aktuelle Ergebnisse ersetzen' (REPLACE) oder 'Nur anhand der Änderungen aktualisieren' (REPLACE_WITH_DIFF) ausgewählt wurde. Während die Spalte RECORD_TIME in Betriebsdatentabellen die Zeit enthält, zu der der Datensatz in die Datenbank eingefügt wurde, enthält diese Spalte in Protokolltabellen die Zeit, zu der der Endpunkt durchsucht wurde (entspricht der Spalte COMPUTER_SCANTIME in der Tabelle COMPUTER).

Um die Protokollfunktion zu aktivieren, müssen Sie im Konfigurations-Repository Protokolltabellen erstellen. Weitere Informationen zur Erstellung dieser Tabellen können Sie dem Abschnitt „Konfigurations-Repository erstellen" auf Seite 63 entnehmen.

Anmerkung: Die Daten werden nie aus den Protokolltabellen gelöscht. Selbst wenn Sie über den Befehl **winvrmnode** ein System aus der TMR entfernen, verbleibt der Datensatz für das entsprechende System in den Protokolltabellen, solange Sie die Protokolltabellen nicht manuell löschen. Weitere Informationen hierzu können Sie dem Abschnitt „Protokolltabellen löschen" auf Seite 368 entnehmen.

Protokolltabellen verwenden

Wenn Sie die Protokollfunktion zum ersten Mal aktivieren, werden im Konfigurations-Repository eine Reihe von Protokolltabellen installiert. Nachdem die Protokolltabellen installiert sind, sollten Sie diese mit Informationen aus einem vollständigen Suchlauf ausfüllen. Wenn Sie das Protokollschema zusammen mit dem Betriebsdatenschema installieren, verwenden Sie die Profiloption 'Nur anhand der Änderungen aktualisieren' für alle Suchläufe. Wenn Sie das Protokollschema nach dem Betriebsdatenschema installieren, verwenden Sie die Profiloption 'Durch aktuelle Ergebnisse ersetzen' für den ersten Suchlauf und 'Nur anhand der Änderungen aktualisieren' für alle weiteren Suchläufe. Dadurch wird gewährleistet, dass nur die geänderten Daten an die Protokolltabellen gesendet werden. Wenn Sie für jeden Suchlauf 'Durch aktuelle Ergebnisse ersetzen' verwenden, werden die

Protokolltabellen bei jedem Suchlauf um die Größe eines vollständigen Suchlaufs vergrößert, was die Abfrage der tatsächlichen Änderungen in der TMR erschwert. Im folgenden Beispiel speichert die Betriebsdatentabelle MATCHED_SWARE Daten zu installierten Softwarepaketen, die über einen Signaturvergleich auf dem Endpunkt gefunden wurden. Beim ersten Suchlauf gibt die Tabelle MATCHED_SWARE 100 Einträge für ein bestimmtes System zurück, so dass die Tabelle H_MATCHED_SWARE ebenfalls 100 Einträge enthält. Vor dem nächsten Suchlauf werden fünf Softwarepakete hinzugefügt und fünf aus dem System entfernt.

Wenn Sie die Option 'Nur anhand der Änderungen aktualisieren' im nächsten Suchlauf verwenden, fügt H_MATCHED_SWARE nur 10 Einträge hinzu, fünf als INSERT-Einträge für die fünf neuen Softwarepakete und fünf als DELETE-Einträge für die fünf Pakete, die entfernt wurden. Wenn Sie jedoch die Option 'Durch aktuelle Ergebnisse ersetzen' verwenden, fügt die Tabelle H_MATCHED_SWARE 100 Einträge hinzu. Zusammen mit den 100 ursprünglichen Einträgen ergibt dies nach dem zweiten Suchlauf insgesamt 200 Einträge, die alle als INSERT-Einträge gespeichert werden. Keiner der Einträge zeigt explizit an, dass einige Softwarepakete gelöscht wurden. Im Gegensatz dazu ergibt die Verwendung der Option 'Nur anhand der Änderungen aktualisieren' insgesamt nur 110 Einträge, und die zusätzlichen Einträge zeigen an, welche Softwarepakete hinzugefügt und welche gelöscht wurden.

Benutzerdefinierte Protokolltabellen

Tivoli Inventory füllt benutzerdefinierte Protokolltabellen automatisch aus, wenn die Tabellen ordnungsgemäß zum Konfigurations-Repository hinzugefügt werden. Weitere Informationen zur Erstellung von benutzerdefinierten Protokolltabellen finden Sie im Abschnitt „Benutzerspezifische Daten mit Tivoli Inventory erfassen" auf Seite 121.

Protokolltabellen löschen

Sie können Protokolltabellen entweder zu einem bestimmten Zeitpunkt oder anhand einer bestimmten Computer-ID manuell löschen. In den folgenden Beispielen wird beschrieben, wie Protokolltabellen für DB2 gelöscht werden können.

Geben Sie folgenden Befehl ein, um Protokolltabellen für ein bestimmtes System zu löschen:

```
delete from H_TABELLENNAME where COMPUTER_SYS_ID = 'SYSTEM-ID';
```

Dabei gilt:

H_TABELLENNAME
: Steht für den Namen der Protokolltabelle.

SYSTEM_ID
: Steht für die Computersystem-ID.

Alle Protokolltabellen mit dem durch *H_TABELLENNAME* angegebenen Namen für das System mit dieser Computersystem-ID werden gelöscht, wenn dieser Befehl ausgeführt wird. Um andere Protokolltabellen für das System mit dieser Computersystem-ID zu löschen, müssen Sie diesen Befehl für jeden Tabellennamen einzeln ausführen.

Geben Sie folgenden Befehl ein, um Protokolltabellen vor einem bestimmten Zeitpunkt zu löschen:

```
delete from H_TABELLENNAME where RECORD_TIME_  < 'JJJJ-MM-TT HH:MM:SS'
```

Dabei gilt:

H_TABELLENNAME
: Steht für den Namen der Protokolltabelle.

JJJJ-MM-TT HH:MM:SS
: Steht für den Zeitpunkt (Datum und Uhrzeit), vor dem Sie Protokolltabellen löschen möchten. Das Format gilt für DB2; andere Plattformen erfordern eventuell andere Formate.

Alle Protokolltabellen mit dem durch *H_TABELLENNAME* angegebenen Namen, die vor diesem Zeitpunkt erstellt wurden, werden gelöscht, wenn dieser Befehl ausgeführt wird.

Ansichten im Konfigurations-Repository

In den folgenden Abschnitten werden Ansichten beschrieben, die standardmäßig in Tivoli Inventory vorhanden sind. Diese Ansichten werden bei der Installation von Tivoli Inventory erstellt. In jedem Abschnitt werden die Spalten der jeweiligen Ansicht beschrieben und die Plattformen angegeben, auf denen die Daten gesammelt werden. Ein Haken (✓) zeigt an, dass die Spalte ausgefüllt werden *kann*, wenn auf der entsprechenden Plattform eine Ansicht ausgeführt wird.

Anmerkung: Ein Haken bedeutet nicht, dass die Spalte jedes Mal ausgefüllt wird, sondern nur, dass die Möglichkeit besteht, dass Daten auf dieser Plattform gesammelt werden. Unter gewissen Umständen kann eine Spalte auch leer bleiben.

Wenn kein Haken angezeigt wird, enthält die Spalte *nie* Daten für die entsprechende Plattform. Ausnahmen werden durch andere Symbole gekennzeichnet und in jedem Abschnitt erklärt.

Anmerkung: Tivoli Inventory Version 4.0 verwendet für alle Datenbanken dieselben Namenskonventionen. Alle Tabellen- und Spaltennamen enthalten höchstens 18 Zeichen und entsprechen dem von RIM verwendeten Abkürzungsschema (siehe „Benutzerspezifische Tabellen, Anzeigen und Spalten für DB2 und Informix benennen” auf Seite 138).

CDROM_VIEW

Zeigt Informationen zu CD-ROM-Laufwerken auf Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓			✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓			✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓			✓	✓	✓
MANUFACTURER	Der Hersteller des installierten CD-ROM-Laufwerks.	✓					✓	✓
MODEL	Das Modell des installierten CD-ROM-Laufwerks.	✓	✓			✓	✓	✓
STORAGE_TYPE	Der Typ des installierten CD-ROM-Laufwerks.	✓	✓			✓	✓	✓
SER_NUM	Die Seriennummer des installierten CD-ROM-Laufwerks.	✓						
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓			✓	✓	✓

COMPUTER_MEM_VIEW

Zeigt Informationen zum Arbeitsspeicher auf Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
PHYSICAL_TOTAL_ KB	Die Gesamtkapazität des installierten Arbeitsspeichers in KB.	✓	✓	✓	✓	✓	✓	✓
PHYSICAL_FREE_ KB	Freier Arbeitsspeicher auf dem System in KB.		✓			✓	✓	✓
TOTAL_PAGES	Die Anzahl der auf dem System verfügbaren Speicher-seiten.	✓	✓		✓	✓	✓	✓
FREE_PAGES	Die Anzahl der freien Speicherseiten auf dem System.	✓	✓			✓	✓	✓
PAGE_SIZE	Die Größe der Seite in Byte.	✓	✓			✓	✓	✓
VIRT_TOTAL_KB	Die gesamte virtuelle Speicherkapazität auf dem System in KB.	✓	✓		✓	✓	✓	✓
VIRT_FREE_KB	Freier virtueller Speicher auf dem System in KB.	✓	✓			✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

COMPUTER_VIEW

Zeigt Informationen zum Computertyp und zum Betriebssystem an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_ MODEL	Informationen zum Systemmodell.	✓	✓			✓	✓	✓
OS_NAME	Das Betriebssystem (z. B. Windows oder Solaris), das zum Zeitpunkt des Suchlaufs aktiv ist.	✓	✓	✓	✓	✓	✓	✓
OS_TYPE	Der Betriebssystemtyp (z. B. Windows NT oder Windows 2000), der zum Zeitpunkt des Suchlaufs aktiv ist.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_ SCANTIME	Der Zeitpunkt des letzten Tivoli Inventory-Suchlaufs.	✓	✓	✓	✓	✓	✓	✓

FLPY_DRV_VIEW

Zeigt Informationen zu Diskettenlaufwerken auf Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓			✓		✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓			✓		✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓			✓		✓	✓
MANUFACTURER	Der Hersteller des installierten Disketten-laufwerks.						W	W
MODEL	Das Modell des installierten Disketten-laufwerks.						W	W
STORAGE_TYPE	Der Typ des installierten Disketten-laufwerks.	✓			✓		✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓			✓		✓	✓

W Nur auf Systemen ***mit*** WMI.

HDISK_VIEW

Zeigt Informationen zu den auf dem System installierten Festplattenlaufwerken an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓		✓	✓	✓	✓
MANUFACTURER	Der Hersteller des installierten Festplattenlaufwerks.	✓					W	
MODEL	Das Modell des installierten Festplattenlaufwerks.	✓	✓			✓	W	✓
STORAGE_TYPE	Der Typ des installierten Festplattenlaufwerks.	✓	✓		✓	✓	✓	✓
SER_NUM	Die Seriennummer des installierten Festplattenlaufwerks.	✓						
HDISK_CYLINDERS	Die Anzahl der Zylinder des Festplattenlaufwerks.	*	*		✓	✓	✓	✓
HDISK_SECTORS	Die Anzahl der Sektoren des Festplattenlaufwerks.	*	*		✓	✓	✓	✓
HDISK_HEADS	Die Anzahl der Köpfe des Festplattenlaufwerks.	*	*		✓	✓	✓	✓
HDISK_SIZE_MB	Die Größe des Festplattenlaufwerks in Megabyte (MB).	✓	✓		✓	✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓		✓	✓	✓	✓

[W] Nur auf Systemen ***mit*** WMI. * Bei AIX-Systemen wird für diesen Wert 0 und bei HP-Systemen -1 gemeldet.

HEADER_INFO_VIEW

Zeigt Header-Informationen zu den auf den Zielsystemen installierten Dateien an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.						✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.						✓	✓
COMPUTER_SYS_ID	Die ID des Systems.						✓	✓
HEADER_NAME	Der Name der Software.						✓	✓
HEADER_VERS	Die Versionsnummer der Software.						✓	✓
HEADER_ PUBLISHER	Der Herausgeber der Software.						✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.						✓	✓

INST_FILE_VIEW

Zeigt Informationen zu den auf den Zielsystemen installierten Dateien an. Diese Ansicht enthält folgende Spalten:

Anmerkung: Auf NetWare-, OS/2- und Windows 98-Systemen können die Spalten CREATED_TIME, MODIFIED_TIME und ACCESSED_TIME abhängig davon ausgefüllt werden, wie viele der Zeitwerte das System unterstützt.

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
FILE_NAME	Der Name der Datei.	✓	✓	✓	✓	✓	✓	✓
FILE_SIZE	Die Größe der Datei.	✓	✓	✓	✓	✓	✓	✓
PATH	Der Pfad für die Datei.	✓	✓	✓	✓	✓	✓	✓
CREATED_TIME	Der Zeitpunkt, zu dem die Datei erstellt wurde.	✓	✓	✓	✓	✓	✓	✓
MODIFIED_TIME	Der Zeitpunkt, zu dem die Datei zuletzt geändert wurde.	✓	✓	✓	✓	✓	✓	✓
ACCESSED_TIME	Der Zeitpunkt, zu dem zuletzt auf die Datei zugegriffen wurde.	✓	✓	✓	✓	✓	✓	✓
FILE_PERMISSIONS	Die Berechtigungen für die Datei.	✓	✓	✓	✓	✓	✓	✓
FILE_OWNER	Der Besitzer der Datei.	✓	✓			✓		
FILE_GROUP	Die Dateigruppe, die die Datei enthält.	✓	✓			✓		

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
CHECKSUM_QUICK	Der 32-Bit-Kontrollsummenwert der schnellen CRC-Prüfung.	✓	✓	✓	✓	✓	✓	✓
CHECKSUM_CRC32	Der 32-Bit-Kontrollsummenwert der vollständigen CRC-Prüfung.	✓	✓	✓	✓	✓	✓	✓
CHECKSUM_MD5	Der 128-Bit-Kontrollsummenwert.	✓	✓	✓	✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

INST_SWARE_VIEW

Zeigt Informationen zu den auf den Zielsystemen installierten Softwarekomponenten an, die über einen Softwaresignaturvergleich auf dem Endpunkt gefunden wurden. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
SWARE_DESC	Die dieser Softwaresignatur zugeordnete Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_VERS	Die Version der dieser Softwaresignatur zugeordneten Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_NAME	Der Name der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
SWARE_SIZE	Die Größe der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

INVENTORYDATA

Zeigt allgemeine Informationen zur Hardware, zum Speicher und zum Betriebssystem der Zielsysteme an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_ MODEL	Das Systemmodell.	✓	✓			✓	✓	✓
PHYSICAL_TOTAL_ KB	Die Gesamtkapazität des installierten Arbeitsspeichers in KB.	✓	✓	✓	✓	✓	✓	✓
PHYSICAL_FREE_ KB	Freier Arbeitsspeicher auf dem System in KB.		✓			✓	✓	✓
TOTAL_PAGES	Die Anzahl der auf dem System verfügbaren Speicherseiten.	✓	✓		✓	✓	✓	✓
FREE_PAGES	Die Anzahl der freien Speicherseiten auf dem System.	✓	✓			✓	✓	✓
PAGE_SIZE	Die Größe der Seite in Byte.	✓	✓			✓	✓	✓
VIRT_TOTAL_KB	Die gesamte virtuelle Speicherkapazität auf dem System.	✓	✓		✓	✓	✓	✓
VIRT_FREE_KB	Freier virtueller Speicher auf dem System.	✓	✓			✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
PROCESSOR_ MODEL	Das Prozessormodell.	✓	✓	✓	✓	✓	✓	✓
PROCESSOR_SPEED	Die aktuelle Geschwindigkeit des Prozessors.	✓	✓	✓	✓	✓	✓	✓
OS_NAME	Das Betriebssystem (z. B. Windows oder Solaris), das zum Zeitpunkt des Suchlaufs aktiv ist.	✓	✓	✓	✓	✓	✓	✓
OS_TYPE	Der Betriebssystemtyp (z. B. Windows NT oder Windows 2000), der zum Zeitpunkt des Suchlaufs aktiv ist.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_ SCANTIME	Der Zeitpunkt des letzten Tivoli Inventory-Suchlaufs.	✓	✓	✓	✓	✓	✓	✓

IP_ADDR_VIEW

Zeigt Informationen zur IP-Adresse (Internet Protocol) auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓		✓	✓	✓	✓
IP_ADDR	Die IP-Adresse für das System.	✓	✓		✓	✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
IP_HOSTNAME	Der IP-Host-Name für das System.	✓	✓			✓	✓	✓
IP_DOMAIN	Der IP-Domänenname für das System.	✓	✓			✓	✓	✓
IP_SUBNET	Das IP-Subnetz für das System.	✓	✓			✓	✓	✓
IP_GATEWAY	Der IP-Gateway-Name für das System.	✓	✓			✓	✓	✓
IP_PRIMARY_DNS	Der erste DNS (Domain Name Service) für das System.	✓	✓			✓	✓	✓
IP_SECONDARY_ DNS	Der zweite DNS für das System.	✓	✓			✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓		✓	✓	✓	✓

IPX_ADDR_VIEW

Zeigt Informationen zur IPX-Adresse auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.			✓			✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.			✓			✓	✓
COMPUTER_SYS_ID	Die ID des Systems.			✓			✓	✓
IPX_ADDR	Die IPX-Adresse des Systems.			✓			✓	✓
NET_NUM	Die Netznummer des Systems.						✓	✓
NODE_ADDR	Die Knotenadresse des Systems.						✓	✓
LINK_SPEED	Die Verbindungs-geschwindigkeit des Systems.						*	*
MAX_PACKET_SIZE	Die maximale Paket-größe, die das System verarbeiten kann.						*	*
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.			✓			✓	✓

* Nicht gemeldet für Systeme, auf denen WMI (Windows Management Interface) aktiv ist.

KEYBOARD_VIEW

Zeigt Informationen zu Tastaturen auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.				✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.				✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.				✓	✓	✓	✓
KEYBOARD_TYPE	Der Typ der an das System angeschlossenen Tastatur.				✓	✓	✓	✓
FUNCTION_KEYS	Die Anzahl der Funktionstasten auf der Tastatur.						✓	✓

MATCH_SWARE_VIEW

Zeigt Informationen zu installierten Softwarekomponenten an, indem nicht abgeglichene Dateidaten mit der Tabelle SWARE_SIG verglichen werden. Während diese Ansicht dieselben Informationen zur Verfügung stellt wie die Ansicht INST_SWARE_VIEW, bei der über Signaturvergleich nach installierter Software gesucht wird, verwendet MATCH_SWARE_VIEW eine ältere Vergleichsmethode. Es wird die Verwendung von INST_SWARE_VIEW empfohlen. MATCH_SWARE_VIEW enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
SWARE_DESC	Die Beschreibung des Softwarepakets.	✓	✓	✓	✓	✓	✓	✓
SWARE_VERS	Die Version des Softwarepakets.	✓	✓	✓	✓	✓	✓	✓
SWARE_NAME	Der Name der diesem Softwarepaket zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
SWARE_SIZE	Die Größe der diesem Softwarepaket zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓

MOUSE_VIEW

Zeigt Informationen zu Mauszeigergeräten auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓			✓		✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓			✓		✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓			✓		✓	✓
BUTTONS	Die Anzahl der Tasten auf der Maus.	✓			✓		✓	✓
MOUSE_MODEL	Das Modell der Maus.						✓	✓
MOUSE_TYPE	Die Art des Zeigegeräts (Maus, Rollkugel usw.).	✓			✓		✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓			✓		✓	✓

NATIV_SWARE_VIEW

Zeigt Produktinformationen zum Betriebssystem auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓			✓	✓	✓
PACKAGE_NAME	Der Name dieses Softwarepakets.	✓	✓			✓	✓	✓
PACKAGE_VERS	Die Version dieses Softwarepakets.	✓	✓			✓	✓	✓
PUBLISHER	Der Herausgeber dieses Softwarepakets.					✓	✓	✓
PACKAGE_ID	Die Paket-ID dieses Softwarepakets.	✓	✓			✓	✓	✓
FILE_PATH	Der Pfad der installierten systemeigenen Software.					✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓			✓	✓	✓

NET_CARD_VIEW

Zeigt Informationen zu den Netzwerkkarten auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓		✓	✓	✓	✓
PERM_MAC_ADDR	Die permanente MAC-Adresse für das System.	✓	✓			✓	✓	✓
CURRENT_ADDR	Die aktuelle Netzwerkadresse für das System.	✓	✓			✓	✓	✓
ADAPTER_TYPE	Der auf dem System installierte Netzwerkadapter.	✓	✓				✓	✓
ADAPTER_MODEL	Das Modell des auf dem System installierten Netzwerkadapters.	✓			✓	✓	✓	✓
MANUFACTURER	Der Hersteller des auf dem System installierten Netzwerkadapters.				✓	✓	✓	✓
INST_DATE	Das Datum, an dem die Netzwerkkarte auf dem System installiert wurde.					W	W	W

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓		✓	✓	✓	✓

[W] Nur auf Systemen ***mit*** WMI.

NOSIG_FILES_VIEW

Zeigt Informationen zu installierten Softwaredateien an, zu denen keine Softwaresignatur passt. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
FILE_NAME	Der Name einer Datei, zu der keine Softwaresignatur passt.	✓	✓	✓	✓	✓	✓	✓
FILE_SIZE	Die Größe einer Datei, zu der keine Softwaresignatur passt.	✓	✓	✓	✓	✓	✓	✓
PATH	Der Pfad einer Datei, zu der keine Softwaresignatur passt.	✓	✓	✓	✓	✓	✓	✓

NW_SERVER_VIEW

Zeigt Informationen zu den Betriebssystemen auf NetWare-Servern an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.			✓				
TME_OBJECT_ID	Die TME-Objekt-ID für das System.			✓				
COMPUTER_SYS_ID	Die ID des Systems.			✓				
NW_DEV_NAME	Der NetWare-Einheitenname auf dem System.			✓				
NW_VERS	Die übergeordnete Versionsnummer der auf dem System installierten NetWare-Software.			✓				
NW_SUB_VERS	Die untergeordnete Versionsnummer der auf dem System installierten NetWare-Software.			✓				
NW_MAX_CONNS	Die maximale Anzahl der auf dem System zulässigen Verbindungen.			✓				
NW_MAX_VOLS	Die maximale Anzahl der auf dem System zulässigen Datenträger.			✓				
NW_REVISION_ LEVEL	Die Revionsstufe der auf dem System installierten NetWare-Software.			✓				

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
NW_SFT_LEVEL	Die auf dem System installierte Version von NetWare SFT (System Fault Tolerant).			✓				
NW_TTS_LEVEL	Die Stufe des auf dem System installierten NetWare-Transaktions-verfolgungssystems (TTS).			✓				
NW_MAX_CONNS_ USED	Die maximale Anzahl der auf dem System verwendeten Verbindungen.			✓				
NW_ACCOUNTING_ VERS	Die auf dem System installierte NetWare-Kontoführungsversion.			✓				
NW_VAP_VERS	Die auf dem System installierte Zusatz-prozessversion (VAP).			✓				
NW_QUEING_VERS	Die auf dem System installierte Version der Warteschlangen-steuerung.			✓				
NW_PRINTSERVR_VERS	Die auf dem System installierte Druckserver-Version.			✓				
NW_VIRT_CONS	Die auf dem System installierte virtuelle Konsole.			✓				
NW_SEC_LEVEL	Die Sicherheits-beschränkungsstufe auf dem System.			✓				

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
NW_INET_BRG_ SUPP	Die auf dem System installierte Internet-Bridge-Unterstützung.			✓				
NW_CLIB_MAJOR_ VERS	Die übergeordnete Versionsnummer der auf dem System installierten C-Laufzeitbibliothek.			✓				
NW_CLIB_MINOR_ VERS	Die untergeordnete Versionsnummer der auf dem System installierten C-Laufzeitbibliothek.			✓				
NW_CLIB_ REVISION	Die Revisionsstufe der auf dem System installierten C-Laufzeitbibliothek.			✓				
NW_SER_NUM	Die Seriennummer der auf dem System installierten NetWare-Software.			✓				
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.			✓				

NW_VOLS_VIEW

Zeigt Informationen zu den NetWare-Datenträgern auf NetWare-Servern an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.			✓				
TME_OBJECT_ID	Die TME-Objekt-ID für das System.			✓				
COMPUTER_SYS_ID	Die ID des Systems.			✓				
NWVOL_NAME	Der Name des auf dem System installierten NetWare-Datenträgers.			✓				
NWVOL_TOTAL_ BLKS	Die Gesamtzahl der Blöcke des auf dem System installierten NetWare-Datenträgers.			✓				
NWVOL_BLK_ SECTORS	Die Anzahl der Sektoren pro Block auf dem NetWare-Datenträger.			✓				
NWVOL_AVAIL_ BLKS	Die Anzahl der verfügbaren Blöcke auf dem NetWare-Datenträger.			✓				
NWVOL_DIR_ SLOTS	Die Gesamtanzahl der auf dem System verfügbaren Verzeichnistabelleneinträge.			✓				
NWVOL_AVAIL_ SLOTS	Die Anzahl der verfügbaren Bereiche auf dem NetWare-Datenträger.			✓				

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
NWVOL_IS_ REMOVABLE	Gibt an, ob der installierte NetWare-Datenträger austauschbar ist.			✓				
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.			✓				

OS_VIEW

Zeigt Informationen zu den Betriebssystemen auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
OS_NAME	Das Betriebssystem (z. B. Windows or Solaris).	✓	✓	✓	✓	✓	✓	✓
OS_TYPE	Der Betriebssystemtyp (z. B. Windows NT oder Windows 2000).	✓	✓	✓	✓	✓	✓	✓
OS_MAJOR_VERS	Die übergeordnete Versionsnummer des Betriebssystems.	✓	✓	✓	✓	✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
OS_MINOR_VERS	Die untergeordnete Versionsnummer des Betriebssystems.	✓	✓	✓	✓	✓	✓	✓
OS_SUB_VERS	Die Unterversions-nummer des Betriebssystems.					✓	✓	✓
OS_INST_DATE	Das Datum, an dem das Betriebssystem installiert wurde.				✓		✓	✓

PARTITION_VIEW

Zeigt Informationen zu den Datenträgerpartitionen auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓		✓	✓	✓	✓
FS_ACCESS_POINT	Der Ort, an dem sich die Partition befindet.	✓	✓		✓	✓	✓	✓
DEV_NAME	Der Gerätename.	✓	✓		✓	✓	✓	✓
PARTITION_TYPE	Die Art der Partition auf dem Laufwerk.	✓	✓		✓	✓	✓	✓
MEDIA_TYPE	Der Datenträgertyp, der die Partition enthält.	✓	✓		✓	✓	✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
PHYSICAL_SIZE_ KB	Die Größe des Laufwerks, das die Partition enthält.	✓	✓		✓	✓	✓	✓
FS_TYPE	Der Dateisystemtyp.	✓	✓		✓	✓	✓	✓
FS_MOUNT_POINT	Der Punkt, an dem die Partition an das Betriebssystem angeschlossen ist.	✓	✓		✓	✓	✓	✓
FS_TOTAL_SIZE_KB	Die Größe der Partition in KB.	✓	✓		✓	✓	✓	✓
FS_FREE_SIZE_KB	Freier Speicherbereich auf der Partition in KB.	✓	✓		✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓		✓	✓	✓	✓

PC_BIOS_VIEW

Zeigt Informationen zum BIOS von PCs an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.						✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.						✓	✓
COMPUTER_SYS_ID	Die ID des Systems.						✓	✓
USER_NAME	Der Benutzername für das System.						✓	✓
DOMAIN_NAME	Der Domänenname für das System.						✓	✓
WORKGROUP_ NAME	Der Arbeitsgruppenname für das System.						✓	[W]
BIOS_ID	Die BIOS-ID für das System.						✓	✓
BIOS_ID_BYTES	Die Größe der BIOS-ID in Byte.						✓	✓
BIOS_DATE	Das Revisionsdatum des BIOS.						✓	✓
BIOS_STRING	Die Beschreibung des BIOS.						✓	✓
BIOS_ MANUFACTURER	Der BIOS-Hersteller.						✓	✓
BIOS_MODEL	Das BIOS-Modell.						✓	✓
BIOS_SER_NUM	Die BIOS-Seriennummer.						✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.						✓	✓

[W] Nur auf Systemen ***mit*** WMI.

PC_PROCESSOR_VIEW

Zeigt Informationen zu Prozessoren auf den Ziel-PCs an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.			✓	✓		✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.			✓	✓		✓	✓
COMPUTER_SYS_ID	Die ID des Systems.			✓	✓		✓	✓
MANUFACTURER	Der Hersteller des Prozessors.			✓	✓		✓	✓
PROCESSOR_ MODEL	Das Modell des Prozessors.			✓	✓		✓	✓
PROCESSOR_SPEED	Die aktuelle Geschwindigkeit des Prozessors.			✓	✓		✓	✓
CHIP_FAMILY	Die Chipfamilie des Prozessors.			✓	✓		✓	✓
CHIP_MODEL	Das Chipmodell des Prozessors.			✓	✓		✓	✓
CHIP_STEPPING	Die Chip Stepping-Einstellung des Prozessors.			✓	✓		✓	✓
VIRT_MODE_EXT	Die Prozessorer-weiterungen für den virtuellen Modus.			✓	✓		✓	✓
PAGE_SIZE_EXT	Die Seitengrößener-weiterungen des Prozessors.			✓	✓		✓	✓
TIME_STAMP_ COUNTER	Der Zeitstempelzähler des Prozessors.			✓	✓		✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
MODEL_ SPECIFIC_REG	Die modellspezifischen Register des Prozessors.			✓	✓		✓	✓
PHYSICAL_ADDR_ EXT	Die physischen Adresserweiterungen des Prozessors.			✓	✓		✓	✓
MACHINECHECK_ EXCPT	Die Maschinen-prüfungsausnahmen des Prozessors.			✓	✓		✓	✓
CMPXCHG8B_SUPP	Die CMPXCHG8B-Anweisungsunter-stützung des Prozessors.			✓	✓		✓	✓
ON_CHIP_APIC	Der integrierte APIC (Advanced Programmable Interrupt Controller) des Prozessors.			✓	✓		✓	✓
MEM_TYPE_ RANGE-_REG	Die Speichertyp-bereichsregister des Prozessors.			✓	✓		✓	✓
PAGE_GLOBAL_ ENABLE	Die PGE-Einstellung (Page Global Enable) für den Prozessor.			✓	✓		✓	✓
MACHINECHECK_ ARCH	Die Maschinen-prüfungsarchitektur des Prozessors.			✓	✓		✓	✓
COND_MOVE_SUPP	Die Prozessorein-stellung für die bedingte Verschiebungsan-weisung.			✓	✓		✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
MMX_ TECHNOLOGY	Die Intel MMX-Funktionen des Prozessors (falls vorhanden).			✓	✓		✓	✓
ON_CHIP_FPU	Die integrierte FPU (Floating Processor Unit) des Prozessors.			✓	✓		✓	✓
DEBUG_EXT_ PRESENT	Gibt an, ob der Prozessor über Debug-Erweiterungen verfügt.			✓	✓		✓	✓
FAST_SYS_CALL	Die Prozessoreinstellung für den schnellen Systemaufruf.			✓	✓		✓	✓
PAGE_ATTR_TABLE	Die Seitenattributtabelle des Prozessors.			✓	✓		✓	✓
PAGE_SIZE_EXT36	Die 36-Bit-Seitengrößenerweiterung des Prozessors.			✓	✓		✓	✓
SER_NUM_ ENABLED	Gibt an, ob die Prozessorseriennummer aktiviert ist.			✓	✓		✓	✓
FAST_FLOAT_SAVE	Die Prozessoreinstellung für schnelles Speichern/Wiederherstellen von Gleitkommata.			✓	✓		✓	✓
SIMD_EXT_SUPP	Die Unterstützung von Streaming SIMD Extensions des Prozessors.			✓	✓		✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
NOW_3_D_ARCH	Die AMD 3DNow!-Funktionen des Prozessors (falls vorhanden).			✓	✓		✓	✓
SER_NUM	Die Seriennummer des Prozessors.			✓	✓		✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.			✓	✓		✓	✓

PCI_DEV_VIEW

Zeigt Informationen zu den auf dem System installierten PCI-Geräten an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.						✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.						✓	✓
COMPUTER_SYS_ID	Die ID des Systems.						✓	✓
INST_PCI_ID	Die PCI-ID des auf dem System installierten oder mit dem System verbundenen PCI-Gerätes.						✓	✓
PCI_DEV_NAME	Der Name des auf dem System installierten oder mit dem System verbundenen PCI-Gerätes.						✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.						✓	✓

PRINTER_VIEW

Zeigt Informationen zu Druckern an, die an die Zielsysteme angeschlossen sind. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.						✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.						✓	✓
COMPUTER_SYS_ID	Die ID des Systems.						✓	✓
PRINTER_MODEL	Das Druckermodell.						✓	✓
PRINTER_NAME	Der Name des Druckers.						✓	✓
PRINTER_ LOCATION	Der Standort des Druckers.						✓	✓
PRINTER_IS_ LOCAL	Gibt an, ob der Drucker ein lokaler Drucker oder ein Netzwerkdrucker ist.						✓	✓
DRV_NAME	Der Name des Druckertreibers für den Drucker.						✓	✓
DRV_VERS	Die Version des Druckertreibers für den Drucker.						✓	✓
PORT_NAME	Der Name des vom Drucker verwendeten Anschlusses.						✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.						✓	✓

PROCESSOR_VIEW

Zeigt Informationen zu den Prozessoren auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
MANUFACTURER	Der Hersteller des Prozessors.		✓	✓	✓		✓	✓
PROCESSOR_ MODEL	Das Prozessormodell.	✓	✓	✓	✓	✓	✓	✓
PROCESSOR_SPEED	Die aktuelle Geschwindigkeit des Prozessors.	✓	✓	✓	✓	✓	✓	✓
SER_NUM	Die Seriennummer dieses Prozessors.			✓	✓		✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

STORAGE_DEV_VIEW

Zeigt Informationen zu den Speichergeräten auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓		✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓		✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓		✓	✓	✓	✓
MANUFACTURER	Der Hersteller des installierten Speichergeräts.	✓					✓	✓
MODEL	Das Modell des installierten Speichergeräts.	✓	✓			✓	✓	✓
STORAGE_TYPE	Der Typ des installierten Speichergeräts.	✓	✓		✓	✓	✓	✓
SER_NUM	Die Seriennummer des installierten Speichergeräts.	✓						
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓		✓	✓	✓	✓

SWARE_MATCH_CRC32

Zeigt auf der Basis von Werten einer vollständigen CRC32-Prüfung (zyklischen Blockprüfung) Informationen zu Dateien an, die auf den Zielsystemen installiert sind. Diese Ansicht vergleicht den CRC32-Kontrollsummenwert aus der Tabelle UNMATCHED_FILES mit dem CRC32-Kontrollsummenwert aus der Tabelle SWARE_SIG. Bevor Sie diese Ansicht ausführen, müssen Sie die Option 'Full' aktivieren und einen Suchlauf durchführen, um grundlegende Informationen zu der entsprechenden Datei zu erhalten. Informationen zur Einstellung des Kontrollsummenwertes finden Sie unter „winvsig" auf Seite 313. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
SWARE_DESC	Die dieser Softwaresignatur zugeordnete Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_VERS	Die Version der dieser Softwaresignatur zugeordneten Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_NAME	Der Name der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
SWARE_SIZE	Die Größe der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

SWARE_MATCH_MD5

Zeigt auf der Basis von MD5-Kontrollsummenwerten Informationen zu Dateien an, die auf den Zielsystemen installiert sind. Diese Ansicht vergleicht den MD5-Kontrollsummenwert aus der Tabelle UNMATCHED_FILES mit dem MD5-Kontrollsummenwert aus der Tabelle SWARE_SIG. Bevor Sie diese Ansicht ausführen, müssen Sie die Option 'MD5' aktivieren und einen Suchlauf durchführen, um grundlegende Informationen zu der entsprechenden Datei zu erhalten. Informationen zur Einstellung des Kontrollsummenwertes finden Sie unter „winvsig" auf Seite 313. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
SWARE_DESC	Die dieser Softwaresignatur zugeordnete Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_VERS	Die Version der dieser Softwaresignatur zugeordneten Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_NAME	Der Name der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
SWARE_SIZE	Die Größe der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

SWARE_MATCH_QUICK

Zeigt auf der Basis von schnellen Kontrollsummenwerten Informationen zu Dateien an, die auf den Zielsystemen installiert sind. Diese Ansicht vergleicht den schnellen Kontrollsummenwert aus der Tabelle UNMATCHED_FILES mit dem schnellen Kontrollsummenwert aus der Tabelle SWARE_SIG. Bevor Sie diese Ansicht ausführen, müssen Sie die Option 'Quick' aktivieren und einen Suchlauf durchführen, um grundlegende Informationen zu der entsprechenden Datei zu erhalten. Informationen zur Einstellung des Kontrollsummenwertes finden Sie unter „winvsig" auf Seite 313. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.	✓	✓	✓	✓	✓	✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.	✓	✓	✓	✓	✓	✓	✓
COMPUTER_SYS_ID	Die ID des Systems.	✓	✓	✓	✓	✓	✓	✓
SWARE_DESC	Die dieser Softwaresignatur zugeordnete Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_VERS	Die Version der dieser Softwaresignatur zugeordneten Software.	✓	✓	✓	✓	✓	✓	✓
SWARE_NAME	Der Name der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
SWARE_SIZE	Die Größe der dieser Softwaresignatur zugeordneten Datei.	✓	✓	✓	✓	✓	✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.	✓	✓	✓	✓	✓	✓	✓

USB_DEV_VIEW

Zeigt Informationen zu USB-Geräten an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.						✓	*
TME_OBJECT_ID	Die TME-Objekt-ID für das System.						✓	*
COMPUTER_SYS_ID	Die ID des Systems.						✓	*
HOST_CNTRL	Der Host-Controller für das USB-Gerät.						✓	*
DEV_ADDR	Die Geräteadresse des USB-Geräts.						✓	*
SER_NUM	Die Seriennummer des USB-Geräts.						✓	*
PORT_NUM	Die Nummer des vom USB-Gerät verwendeten Anschlusses.						✓	*
PARENT_ADDR	Die vom USB-Gerät verwendete übergeordnete Adresse.						✓	*
USB_VERS	Die Version des USB-Geräts (z. B. USB 1.1 oder USB 2.0).						✓	*
DEV_CLASS	Die Geräteklasse des USB-Geräts.						✓	*
DEV_SUBCLASS	Die Geräteunterklasse des USB-Geräts.						✓	*
VENDOR_ID	Die Herstellerkennung des USB-Geräts.						✓	*

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
PRODUCT_ID	Die Produkt-ID des USB-Geräts.						✓	*
MANUFACTURER	Der Hersteller des USB-Geräts.						✓	*
PRODUCT	Das USB-Produkt.						✓	*
NUM_OF_PORTS	Die Anzahl der USB-Anschlüsse am USB-Gerät.						✓	*
DEV_IS_HUB	Gibt an, ob das USB-Gerät ein USB-Hub ist.						✓	*
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.						✓	*
* Windows NT unterstützt keine USB-Geräte.								

VID_CARD_VIEW

Zeigt Informationen zu Videokarten auf den Zielsystemen an. Diese Ansicht enthält folgende Spalten:

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
TME_OBJECT_ LABEL	Die TME-Objektbezeichnung für das System.				✓		✓	✓
TME_OBJECT_ID	Die TME-Objekt-ID für das System.				✓		✓	✓
COMPUTER_SYS_ID	Die ID des Systems.				✓		✓	✓
VID_CARD_MODEL	Der Hersteller der Videokarte.				✓		✓	✓

Spaltenname	Beschreibung	AIX	GB	NetWare	OS/2	Solaris	Win 98	Win NT/2000
VID_CARD_BIOS	Die BIOS-Informationen für die Videokarte.						✓	✓
VID_DAC_TYPE	Der integrierte Digital-Analog-Umsetzer (DAC) der Videokarte.						✓	✓
VID_MEM	Die Speicherkapazität der Videokarte.				✓		✓	✓
VID_BIOS_ RELDATE	Das Freigabedatum des BIOS der Videokarte.						✓	✓
VID_CHIP_TYPE	Der Chiptyp der Videokarte.						✓	✓
VID_HORIZNTL_ RES	Die Einstellung für die horizontale Auflösung der installierten Videokarte.						✓	✓
VID_VERTICAL_ RES	Die Einstellung für die vertikale Auflösung der installierten Videokarte.						✓	✓
VID_COLORS	Die Farbeinstellung der installierten Videokarte.						✓	✓
RECORD_TIME	Der Zeitpunkt, zu dem die Daten in die Datenbank eingefügt wurden.				✓		✓	✓

Protokollansichten im Konfigurations-Repository

Im Allgemeinen bestehen die Namen der Protokollansichten des Konfigurations-Repositorys aus 'H_' am Anfang des Namens und dem Namen der Nicht-Protokollansicht, auf der die entsprechende Protokollansicht basiert. Eine Protokollansicht gibt alle Spalten zurück, die auch die entsprechende Nicht-Protokollansicht zurückgibt, sowie zusätzlich die beiden folgenden Spalten:

Spaltenname	Beschreibung
RECORD_ACTION	Diese Spalte kann folgende Werte enthalten: INSERT (neue Informationen werden zur Betriebsdatentabelle hinzugefügt), UPDATE (ein Datensatz in der Betriebsdatentabelle wird teilweise geändert) oder DELETE (der Datensatz ist in der Betriebsdatentabelle nicht mehr vorhanden).
PRFL_ACTION	Gibt an, ob als Profilkonfigurationsoption REPLACE (Durch aktuelle Ergebnisse ersetzen) oder REPLACE_WITH_DIFF (Nur anhand der Änderungen aktualisieren) ausgewählt wurde.

In der folgenden Liste werden die verfügbaren Protokollansichten aufgeführt:

H_CDROM_VIEW

H_COMPUTER_VIEW

H_FLPY_DRV_VIEW

H_HDISK_VIEW

H_HEADER_VIEW

H_INST_FILE_VIEW

H_INST_SWARE_VIEW

H_IP_ADDR_VIEW

H_IPX_ADDR_VIEW

H_MEM_VIEW

H_MOUSE_VIEW

H_NATIV_VIEW

H_NET_CARD_VIEW

H_NOSIG_FILES_VIEW

H_NW_SERVER_VIEW

H_NW_VOLS_VIEW

H_OS_VIEW

H_PARTITION_VIEW

H_PC_BIOS_VIEW

H_PCI_DEV_VIEW

H_PCPROCESSOR_VIEW

H_PRINTER_VIEW

H_PROCESSOR_VIEW

H_STORAGE_DEV_VIEW

H_TAPEDRV_VIEW

H_UNIX_SYS_VIEW

H_USB_DEV_VIEW

H_VID_CARD_VIEW

Eine ausführliche Beschreibung der Ansichten und der mit ihnen zurückgegebenen Spalten finden Sie unter den entsprechenden Nicht-Protokollansichten.

Tivoli Inventory- und Subskriptionsabfragen

In den folgenden Abschnitten werden die standardmäßig in Tivoli Inventory vorhandenen Abfragen sowie deren Spalten aufgeführt. Anweisungen zur Erstellung von Tivoli Inventory-Abfragen finden Sie im Kapitel „Tivoli Inventory Version 4.0 installieren” auf Seite 19. Anweisungen zur Erstellung eigener Abfragen finden Sie im Kapitel „Inventory-Daten abfragen” auf Seite 209.

Tivoli Inventory-Abfragen

In der folgenden Übersicht werden die standardmäßig in Tivoli Inventory vorhandenen Abfragen beschrieben und deren Spalten aufgeführt. Beschreibungen der Spaltennamen finden Sie unter der Ansicht, an die sich die Abfrage richtet.

CDROM_QUERY

Von dieser Abfrage werden Angaben zum Modell und zu physischen Details der auf den Systemen installierten

CD-ROM-Laufwerke zurückgegeben. Die Abfrage wird für die Ansicht CDROM_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

MODEL

STORAGE_TYPE

SER_NUM

RECORD_TIME

COMPUTER_MEM_QUERY

Von dieser Abfrage werden Angaben zu dem auf einem System installierten Hauptspeicher zurückgegeben. Die Abfrage wird für die Ansicht COMPUTER_MEM_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

PHYSICAL_TOTAL_KB

PHYSICAL_FREE_KB

TOTAL_PAGES

FREE_PAGES

PAGE_SIZE

VIRT_TOTAL_KB

VIRT_FREE_KB

RECORD_TIME

FLPY_DRV_QUERY

Von dieser Abfrage werden Angaben zum Modell und zu

physischen Details der auf den Systemen installierten Diskettenlaufwerke zurückgegeben. Die Abfrage wird für die Ansicht FLPY_DRV_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

MODEL

STORAGE_TYPE

RECORD_TIME

HDISK_QUERY

Von dieser Abfrage werden Angaben zum Modell und zu physischen Details der auf den Systemen installierten Festplatten zurückgegeben. Die Abfrage wird für die Ansicht HDISK_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

MODEL

STORAGE_TYPE

SER_NUM

HDISK_CYLINDERS

HDISK_SECTORS

HDISK_HEADS

HDISK_SIZE_MB

RECORD_TIME

HEADER_INFO_QUERY

Von dieser Abfrage werden allgemeine Angaben zum Software-Header zurückgegeben. Die Abfrage wird für die Ansicht HEADER_INFO_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

HEADER_NAME

HEADER_VERS

HEADER_PUBLISHER

RECORD_TIME

INST_FILE_QUERY

Anmerkung: Diese Abfrage ist sehr groß, so dass ihre Ausführung lange dauert.

Von dieser Abfrage werden Angaben zu den auf einem System installierten Dateien zurückgegeben. Die Abfrage wird für die Ansicht INST_FILE_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

FILE_NAME

FILE_SIZE

PATH

CREATED_TIME

MODIFIED_TIME

ACCESSED_TIME

FILE_PERMISSIONS

FILE_OWNER

FILE_GROUP

CHECKSUM_QUICK

CHECKSUM_CRC32

CHECKSUM_MD5

RECORD_TIME

INVENTORY_HWARE

Von dieser Abfrage werden allgemeine Angaben zur Systemhardware zurückgegeben. Die Abfrage wird für die Ansicht INVENTORYDATA ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

COMPUTER_SCANTIME

COMPUTER_MODEL

OS_NAME

OS_TYPE

PROCESSOR_MODEL

PROCESSOR_SPEED

PHYSICAL_TOTAL_KB

PHYSICAL_FREE_KB

TOTAL_PAGES

FREE_PAGES

PAGE_SIZE

VIRT_TOTAL_KB

VIRT_FREE_KB

INVENTORY_SWARE

Von dieser Abfrage werden allgemeine Angaben zur Software zurückgegeben. Die Abfrage wird für die Ansicht INST_SWARE_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

SWARE_DESC

SWARE_VERS

SWARE_NAME

SWARE_SIZE

RECORD_TIME

IP_ADDR_QUERY

Von dieser Abfrage werden Angaben zur IP-Adresse eines Systems zurückgegeben. Die Abfrage wird für die Ansicht IP_ADDR_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

IP_ADDR

IP_HOSTNAME

IP_DOMAIN

IP_SUBNET

IP_GATEWAY

IP_PRIMARY_DNS

IP_SECONDARY_DNS

RECORD_TIME

IPX_ADDR_QUERY

Von dieser Abfrage werden Angaben zur IPX-Adresse eines Systems zurückgegeben. Die Abfrage wird für die Ansicht IPX_ADDR_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

IPX_ADDR

NET_NUM

NODE_ADDR

LINK_SPEED

MAX_PACKET_SIZE

RECORD_TIME

KEYBOARD_QUERY

Von dieser Abfrage werden Angaben zu der auf einem System installierten Tastatur zurückgegeben. Die Abfrage wird für die Ansicht KEYBOARD_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

KEYBOARD_TYPE

FUNCTION_KEYS

MATCH_CRC32_QUERY

Anmerkung: Diese Abfrage ist sehr groß, so dass ihre Ausführung lange dauert.

Von dieser Abfrage werden allgemeine Angaben zur Software zurückgegeben, die bei einem auf einer CRC32-Kontrollsummenprüfung basierenden Softwaresignaturvergleich gefunden wird. Die Abfrage wird für die Ansicht SWARE_MATCH_CRC32 ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

SWARE_DESC

SWARE_VERS

SWARE_NAME

SWARE_SIZE

CHECKSUM_CRC32

RECORD_TIME

MATCH_MD5_QUERY

Anmerkung: Diese Abfrage ist sehr groß, so dass ihre Ausführung lange dauert.

Von dieser Abfrage werden allgemeine Angaben zur Software zurückgegeben, die bei einem auf einer MD5-Kontrollsummenprüfung basierenden Softwaresignatur-

vergleich gefunden wird. Die Abfrage wird für die Ansicht SWARE_MATCH_MD5 ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

SWARE_DESC

SWARE_VERS

SWARE_NAME

SWARE_SIZE

CHECKSUM_MD5

RECORD_TIME

MATCH_QUICK_QUERY

Anmerkung: Diese Abfrage ist sehr groß, so dass ihre Ausführung lange dauert.

Von dieser Abfrage werden allgemeine Angaben zur Software zurückgegeben, die bei einem auf einer schnellen Kontrollsummenprüfung basierenden Softwaresignaturvergleich gefunden wird. Die Abfrage wird für die Ansicht SWARE_MATCH_QUICK_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

SWARE_DESC

SWARE_VERS

SWARE_NAME

SWARE_SIZE

CHECKSUM_QUICK

RECORD_TIME

MOUSE_QUERY

Von dieser Abfrage werden Angaben zu der auf einem System installierten Maus zurückgegeben. Die Abfrage wird für die Ansicht MOUSE_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

BUTTONS

MOUSE_MODEL

MOUSE_TYPE

RECORD_TIME

NATIV_SWARE_QUERY

Von dieser Abfrage werden allgemeine Angaben zur Software zurückgegeben. Die Abfrage wird für die Ansicht NATIV_SWARE_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

PACKAGE_NAME

PACKAGE_VERS

PUBLISHER

PACKAGE_ID

RECORD_TIME

NET_CARD_QUERY

Von dieser Abfrage werden Angaben zu den auf den Systemen installierten Netzwerkkarten zurückgegeben. Die Abfrage wird für die Ansicht NET_CARD_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

PERM_MAC_ADDR

CURRENT_ADDR

ADAPTER_TYPE

ADAPTER_MODEL

MANUFACTURER

INST_DATE

NOSIG_FILES_QUERY

Anmerkung: Diese Abfrage ist sehr groß, so dass ihre Ausführung lange dauert.

Von dieser Abfrage werden allgemeine Angaben zu Dateien zurückgegeben, zu denen keine Softwaresignatur passt. Die Abfrage wird für die Ansicht NOSIG_FILES_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

FILE_NAME

FILE_SIZE

NW_SERVER_QUERY

Von dieser Abfrage werden Angaben zu den auf NetWare-Servern installierten Betriebssystemen zurückgegeben. Die Abfrage wird für die Ansicht NW_SERVER_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

NW_DEV_NAME

NW_VERS

NW_SUB_VERS

NW_MAX_CONNS

NW_MAX_VOLS

NW_REVISION_LEVEL

NW_SFT_LEVEL

NW_TTS_LEVEL

NW_MAX_CONNS_USED

NW_ACCOUNTING_VERS

NW_VAP_VERS

NW_QUEING_VERS

NW_PRINTSERVR_VERS

NW_VIRT_CONS

NW_SEC_LEVEL

NW_INET_BRG_SUPP

NW_CLIB_MAJOR_VERS

NW_CLIB_MINOR_VERS

NW_CLIB_REVISION

NW_SER_NUM

RECORD_TIME

NW_VOLS_QUERY

Von dieser Abfrage werden Angaben zu NetWare-Datenträgern für NetWare-Systeme zurückgegeben. Die Abfrage wird für die Ansicht NW_VOLS_DET_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

NWVOL_NAME

NWVOL_TOTAL_BLKS

NWVOL_BLK_SECTORS

NWVOL_AVAIL_BLKS

NWVOL_DIR_SLOTS

NWVOL_AVAIL_SLOTS

NWVOL_IS_REMOVABLE

RECORD_TIME

OS_QUERY

Von dieser Abfrage werden Angaben zu einem Betriebssystem zurückgegeben. Die Abfrage wird für die Ansicht OS_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

OS_NAME

OS_TYPE

OS_MAJOR_VERS

OS_MINOR_VERS

OS_SUB_VERS

OS_INST_DATE

PARTITION_QUERY

Von dieser Abfrage werden Angaben zu den Festplattenpartitionen auf den Systemen zurückgegeben. Die Abfrage wird für die Ansicht PARTITION_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

FS_ACCESS_POINT

DEV_NAME

PARTITION_TYPE

MEDIA_TYPE

PHYSICAL_SIZE_KB

FS_TYPE

FS_MOUNT_POINT

FS_TOTAL_SIZE_KB

FS_FREE_SIZE_KB

RECORD_TIME

PC_BIOS_QUERY

Von dieser Abfrage werden die verfügbaren Angaben zum BIOS auf einem PC-System zurückgegeben. Die Abfrage wird für die Ansicht PC_BIOS_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

USER_NAME

DOMAIN_NAME

WORKGROUP_NAME

BIOS_ID

BIOS_ID_BYTES

BIOS_DATE

BIOS_STRING

BIOS_MANUFACTURER

BIOS_MODEL

BIOS_SER_NUM

RECORD_TIME

PC_PROCESSOR_QUERY

Von dieser Abfrage werden Angaben zu dem auf einem PC-System installierten Prozessortyp zurückgegeben. Die Abfrage wird für die Ansicht PC_PROCESSOR_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

PROCESSOR_MODEL

PROCESSOR_SPEED

SER_NUM

CHIP_FAMILY

CHIP_MODEL

CHIP_STEPPING

VIRT_MODE_EXT

PAGE_SIZE_EXT

TIME_STAMP_COUNTER

MODEL_SPECIFIC_REG

PHYSICAL_ADDR_EXT

MACHINECHECK_EXCPT

CMPXCHG8B_SUPP

ON_CHIP_APIC

MEM_TYPE_RANGE_REG

PAGE_GLOBAL_ENABLE

MACHINECHECK_ARCH

COND_MOVE_SUPP

MMX_TECHNOLOGY

ON_CHIP_FPU

DEBUG_EXT_PRESENT

FAST_SYS_CALL

PAGE_ATTR_TABLE

PAGE_SIZE_EXT36

SER_NUM_ENABLED

FAST_FLOAT_SAVE

SIMD_EXT_SUPP

NOW_3_D_ARCH

RECORD_TIME

PCI_DEV_QUERY

Von dieser Abfrage werden Angaben zu den auf einem System installierten PCI-Geräten zurückgegeben. Die Abfrage wird für die Ansicht PCI_DEV_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

INST_PCI_ID

PCI_DEV_NAME

RECORD_TIME

PRINTER_QUERY

Von dieser Abfrage werden Angaben zu dem an ein System angeschlossenen Drucker zurückgegeben. Die Abfrage wird für die Ansicht PRINTER_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

PRINTER_MODEL

PRINTER_NAME

PRINTER_LOCATION

PRINTER_IS_LOCAL

DRV_NAME

DRV_VERS

PORT_NAME

RECORD_TIME

PROCESSOR_QUERY

Von dieser Abfrage werden Angaben zu dem auf einem System installierten Prozessormodell zurückgegeben. Die Abfrage wird für die Ansicht PROCESSOR_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

PROCESSOR_MODEL

PROCESSOR_SPEED

SER_NUM

RECORD_TIME

STORAGE_DEV_QUERY

Von dieser Abfrage werden Speicherangaben für ein System zurückgegeben. Die Abfrage wird für die Ansicht STORAGE_DEV_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

MODEL

STORAGE_TYPE

SER_NUM

RECORD_TIME

TAPEDRV_QUERY

Von dieser Abfrage werden Angaben zum Modell und zu physischen Details der auf den Systemen installierten Bandlaufwerke zurückgegeben. Die Abfrage wird für die Ansicht TAPEDRV_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

MANUFACTURER

MODEL

STORAGE_TYPE

SER_NUM

RECORD_TIME

UNIX_SYS_QUERY

Von dieser Abfrage werden die verfügbaren Angaben zu einem UNIX-System zurückgegeben. Die Abfrage wird für die Ansicht UNIX_SYS_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

BOOT_TIME

UPTIME

RUN_LEVEL

HOST_NAME

RECORD_TIME

USB_DEV_QUERY

Von dieser Abfrage werden Angaben zu den an ein System angeschlossenen USB-Geräten zurückgegeben. Die Abfrage wird für die Ansicht USB_DEV_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

HOST_CNTRL

DEV_ADDR

SER_NUM

PORT_NUM

PARENT_ADDR

USB_VERS

DEV_CLASS

DEV_SUBCLASS

VENDOR_ID

PRODUCT_ID

MANUFACTURER

PRODUCT

NUM_OF_PORTS

DEV_IS_HUB

RECORD_TIME

VID_CARD_QUERY

Von dieser Abfrage werden Angaben zu der auf einem System installierten Videokarte zurückgegeben. Die Abfrage wird für die Ansicht VID_CARD_VIEW ausgeführt und gibt die folgenden Spalten zurück:

TME_OBJECT_LABEL

TME_OBJECT_ID

COMPUTER_SYS_ID

VID_CARD_MODEL

VID_CARD_BIOS

VID_DAC_TYPE

VID_MEM

VID_BIOS_RELDATE

VID_CHIP_TYPE

VID_HORIZNTL_RES

VID_VERTICAL_RES

VID_COLORS

RECORD_TIME

Tivoli Inventory-Protokollabfragen

Die Namen der Inventory-Protokollabfragen bestehen aus 'H_' und dem Namen der Nicht-Protokollabfrage, auf der die entsprechende Protokollabfrage basiert. Eine Protokollabfrage gibt alle Spalten zurück, die auch die entsprechende Nicht-Protokollabfrage zurückgibt, sowie zusätzlich die Spalten RECORD_ACTION und PRFL_ACTION.

In der folgenden Liste werden die verfügbaren Protokollabfragen aufgeführt:

H_CDROM_QUERY

H_COMPUTER_MEM_QUERY

H_COMPUTER_QUERY

H_FLPY_DRV_QUERY

H_HDISK_QUERY

H_HEADER_INFO_QUERY

H_INST_FILE_QUERY

Anmerkung: H_INST_FILE_QUERY ist sehr groß, so dass ihre Ausführung lange dauert.

H_INVENTORY_SWARE

H_IP_ADDR_QUERY

H_IPX_ADDR_QUERY

H_MOUSE_QUERY

H_NATIV_SWARE_QUERY

H_NET_CARD_QUERY

H_NOSIG_FILES_QUERY

Anmerkung: H_NOSIG_FILES_QUERY ist sehr groß, so dass ihre Ausführung lange dauert.

H_NW_SERVER_QUERY

H_NW_VOLS_QUERY

H_OS_QUERY

H_PARTITION_QUERY

H_PC_BIOS_QUERY

H_PCI_DEV_QUERY

H_PCPROCESSOR_QUERY

H_PRINTER_QUERY

H_PROCESSOR_QUERY

H_STORAGE_DEV_QUERY

H_TAPEDRV_QUERY

H_UNIX_SYS_QUERY

H_USB_DEV_QUERY

H_VID_CARD_QUERY

Subskriptionsabfragen

Im Folgenden finden Sie eine Übersicht über die standardmäßig in Tivoli Inventory vorhandenen Subskriptionsabfragen. Weitere Informationen zu Subskriptionsabfragen sowie zur Erstellung eigener Abfragen können Sie dem Kapitel „Inventory-Daten abfragen” auf Seite 209 entnehmen.

Von den folgenden Abfragen werden die Namen der Systeme zurückgegeben, die über das jeweilige Betriebssystem verfügen:

Subskriptionsabfrage	Beschreibung	Ausgeführt für Ansicht
SUB_128MB_ SUBSCRIPTION	Abfrage für Systeme mit höchstens 128 MB Arbeitsspeicher.	INVENTORY DATA
128MB_SUBSCRIPTION	Abfrage für Systeme mit mehr als 128 MB Arbeitsspeicher.	INVENTORY DATA
256MB_SUBSCRIPTION	Abfrage für Systeme mit mehr als 256 MB Arbeitsspeicher.	INVENTORY DATA
512MB_SUBSCRIPTION	Abfrage für Systeme mit mehr als 512 MB Arbeitsspeicher.	INVENTORY DATA
AIX_SUBSCRIPTION	Abfrage für alle AIX-Systeme.	COMPUTER_ VIEW
HPUX_SUBSCRIPTION	Abfrage für alle HP-UX-Systeme.	COMPUTER_ VIEW
NETWARE_ SUBSCRIPTION	Abfrage für alle NetWare-Systeme.	COMPUTER_ VIEW
OS2_SUBSCRIPTION	Abfrage für alle OS/2-Systeme.	COMPUTER_ VIEW
Solaris_SUBSCRIPTION	Abfrage für alle Solaris-Systeme.	COMPUTER_ VIEW
WIN_2000_ SUBSCRIPTION	Abfrage für alle Windows 2000-Systeme.	COMPUTER_ VIEW
WIN_98_SUBSCRIPTION	Abfrage für alle Windows 98-Systeme.	COMPUTER_ VIEW
WIN_ALL_ SUBSCRIPTION	Abfrage für alle Windows-Systeme.	COMPUTER_ VIEW
WIN_ME_ SUBSCRIPTION	Abfrage für alle Windows Me-Systeme.	COMPUTER_ VIEW

Subskriptionsabfrage	Beschreibung	Ausgeführt für Ansicht
WIN_NT_ SUBSCRIPTION	Abfrage für alle Windows NT-Systeme.	COMPUTER_ VIEW

Alle diese Abfragen geben die folgenden Spalten zurück:

TME_OBJECT_ID

TME_OBJECT_LABEL

Protokollsubskriptionsabfragen

Im Folgenden finden Sie eine Übersicht über die standardmäßig in Tivoli Inventory vorhandenen Protokollsubskriptionsabfragen. Weitere Informationen zu Subskriptionsabfragen sowie zur Erstellung eigener Abfragen können Sie dem Kapitel „Inventory-Daten abfragen” auf Seite 209 entnehmen.

Von den folgenden Protokollsubskriptionsabfragen werden die Namen der Systeme zurückgegeben, die über das jeweilige Betriebssystem verfügen:

Subskriptionsabfrage	Beschreibung	Ausgeführt in Ansicht
H_AIX_SUBSCRIPTION	Protokollabfrage für alle AIX-Systeme.	COMPUTER_ VIEW
H_HPUX_ SUBSCRIPTION	Protokollabfrage für alle HP-UX-Systeme.	COMPUTER_ VIEW
H_NETWARE_ SUBSCRIPTION	Protokollabfrage für alle NetWare-Systeme.	COMPUTER_ VIEW
H_OS2_SUBSCRIPTION	Protokollabfrage für alle OS/2-Systeme.	COMPUTER_ VIEW
H_Solaris_ SUBSCRIPTION	Protokollabfrage für alle Solaris-Systeme.	COMPUTER_ VIEW
H_WIN_2000_ SUBSCRIPTION	Protokollabfrage für alle Windows 2000-Systeme.	COMPUTER_ VIEW

Subskriptionsabfrage	Beschreibung	Ausgeführt in Ansicht
H_WIN_98_ SUBSCRIPTION	Protokollabfrage für alle Windows 98-Systeme.	COMPUTER_ VIEW
H_WIN_ALL_ SUBSCRIPTION	Protokollabfrage für alle Windows-Systeme.	COMPUTER_ VIEW
H_WIN_ME_ SUBSCRIPTION	Protokollabfrage für alle Windows Me-Systeme.	COMPUTER_ VIEW
H_WIN_NT_ SUBSCRIPTION	Protokollabfrage für alle Windows NT-Systeme.	COMPUTER_ VIEW

Alle Protokollsubskriptionsabfragen geben die folgenden Spalten zurück:

TME_OBJECT_ID

TME_OBJECT_LABEL

Tabellen im Konfigurations-Repository

In diesem Abschnitt werden die Betriebsdatentabellen im Konfigurations-Repository, die von Tivoli Inventory ausgefüllt werden, beschrieben und deren Spalten aufgeführt.

Anmerkung: Die Spalten werden unter „Ansichten im Konfigurations-Repository” auf Seite 370 beschrieben. Die Beschreibung einer Spalte finden Sie am besten über den entsprechenden Spaltennamen im Index dieses Handbuchs.

Einige Tabellen im Konfigurations-Repository werden nicht durch Tivoli Inventory ausgefüllt. Diese Tabellen sind für künftige Erweiterungen reserviert, daher sollten Sie hier auch keine Benutzerdaten ablegen. Anweisungen zum Hinzufügen von Tabellen im Konfi-

gurations-Repository können Sie dem Kapitel „Benutzerspezifische Daten mit Tivoli Inventory erfassen” auf Seite 121 entnehmen.

COMPUTER

Beschreibung

Enthält allgemeine Angaben zu einem Computersystem. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

COMPUTER_SCANTIME

TME_OBJECT_ID

TME_OBJECT_LABEL

COMPUTER_MODEL

COMPUTER_BOOT_TIME

COMPUTER_ALIAS

SYS_SER_NUM

OS_NAME

OS_TYPE

OS_MAJOR_VERS

OS_MINOR_VERS

OS_SUB_VERS

OS_INST_DATE

REGISTERED_OWNER

REGISTERED_ORG

KEYBOARD_TYPE

FUNCTION_KEYS

TZ_LOCALE

TZ_NAME

TZ_DAYLIGHT_NAME

ON_SAVINGS_TIME

TZ_SECONDS

TIME_DIRECTION

RECORD_TIME

COMPUTER_SYS_MEM

Beschreibung

Enthält Angaben zu dem auf einem System installierten physischen und virtuellen Hauptspeicher. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

PHYSICAL_TOTAL_KB

PHYSICAL_FREE_KB

TOTAL_PAGES

FREE_PAGES

PAGE_SIZE

VIRT_TOTAL_KB

VIRT_FREE_KB

RECORD_TIME

FILE_DESC

Beschreibung

Enthält den Namen und die Größe einer Datei, die bei einer Suche nach grundlegenden Dateiinformationen gefunden

wurde; wird durch die Tabelle UNMATCHED_FILES ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für eine bestimmte Datei, auch wenn genau die gleiche Datei auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

FILE_DESC_ID

FILE_NAME

FILE_SIZE

FILE_FILTER

Beschreibung

Enthält den Dateinamen eines Dateifilters. Diese Tabelle wird von Software-Suchprogrammen zur Filterung verwendet.

Anmerkung

Wird durch den Befehl **winvfilter** und über die GUI für Tivoli Inventory Version 4.0 ausgefüllt.

Spalten

FILE_NAME

FILE_PATH

Beschreibung

Enthält den Pfad für eine Datei, die bei einer Suche nach grundlegenden Dateiinformationen gefunden wurde; wird durch die Tabelle UNMATCHED_FILES ergänzt. Diese Tabelle wird durch einen Suchlauf nach grundlegenden Dateiinformationen ausgefüllt. Das Konfigurations-Repository enthält nur eine Tabelle für eine bestimmte Datei, auch wenn genau dieselbe Datei auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

FILE_PATH_ID

PATH

HDISK

Beschreibung

Enthält Angaben zu einem bestimmten Festplattentyp- und modell; wird durch die Tabelle STORAGE_DEV ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für ein bestimmtes Gerät, auch wenn genau das gleiche Gerät auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

HDISK_ID

HDISK_CYLINDERS

HDISK_SECTORS

HDISK_HEADS

HDISK_SIZE_MB

HEADER_INFO

Beschreibung

Enthält Angaben zu einem Software-Header; wird durch die Tabelle INST_HEADER_INFO ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für eine bestimmte Datei, auch wenn genau die gleiche Datei auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf (Windows) ausgefüllt.

Spalten

HEADER_ID

HEADER_NAME

HEADER_VERS

HEADER_PUBLISHER

INST_HEADER_INFO

Beschreibung

Enthält den Speicherort der Tabelle HEADER_INFO, die Angaben zur Datei enthält. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf (Windows) ausgefüllt.

Spalten

COMPUTER_SYS_ID

HEADER_ID

RECORD_TIME

INST_MOUSE

Beschreibung

Enthält Angaben zum Typ der installierten Maus, zu den Betriebssystemeinstellungen und zum Speicherort der Tabelle MOUSE, die weitere Angaben zur Maus enthält. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

MOUSE_ID

INST_MOUSE_ID

RECORD_TIME

INST_NATIV_SWARE

Beschreibung

Enthält den Dateipfad und den Speicherort der Tabelle NATIV_SWARE, die Angaben zur systemeigenen Software enthält. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

NATIV_ID

FILE_PATH

RECORD_TIME

INST_PARTITION

Beschreibung

Enthält Angaben zu einer Datenträgerpartition auf einem Laufwerk des Systems. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

FS_ACCESS_POINT

DEV_NAME

PARTITION_TYPE

MEDIA_TYPE

PHYSICAL_SIZE_KB

FS_TYPE

FS_MOUNT_POINT

FS_TOTAL_SIZE_KB

FS_FREE_SIZE_KB

RECORD_TIME

INST_PRINTER

Beschreibung

Enthält den Typ des installierten Druckers, die Treibersoftware und die Anschlusseinstellungen. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

PRINTER_ID

INST_PRINTER_ID

PRINTER_NAME

PRINTER_LOCATION

PRINTER_IS_LOCAL

DRV_NAME

DRV_VERS

PORT_NAME

RECORD_TIME

INST_PROCESSOR

Beschreibung

Enthält Angaben zum Typ der installierten Prozessoren. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

PROCESSOR_NUM

PROCESSOR_ID

SER_NUM

RECORD_TIME

INST_USB_DEV

Beschreibung

Enthält Angaben zum Typ des installierten USB-Geräts. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

USB_ID

HOST_CNTRL

DEV_ADDR

SER_NUM

PORT_NUM

PARENT_ADDR

RECORD_TIME

INST_VID_CARD

Beschreibung

Enthält Angaben zum Typ der installierten Videokarte und zu den Betriebssystemeinstellungen. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

VID_CARD_ID

INST_VID_CARD_ID

VID_HORIZNTL_RES

VID_VERTICAL_RES

VID_COLORS

RECORD_TIME

IP_ADDR

Beschreibung

Enthält Angaben zur IP-Adresse (Internet Protocol) des Systems. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

IP_ADDR

IP_HOSTNAME

IP_DOMAIN

IP_SUBNET

IP_GATEWAY

IP_PRIMARY_DNS

IP_SECONDARY_DNS

RECORD_TIME

IPX_ADDR

Beschreibung

Enthält Angaben zur IPX-Adresse des Systems. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

IPX_ADDR

NET_NUM

NODE_ADDR

LINK_SPEED

MAX_PACKET_SIZE

RECORD_TIME

MATCHED_SWARE

Beschreibung

Enthält Angaben zu einer Datei, zu der eine Softwaresignatur passt. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

SWARE_SIG_ID

RECORD_TIME

MOUSE

Beschreibung

Enthält Angaben zu einem bestimmten Maustyp- und modell;

wird durch die Tabelle INST_MOUSE ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für ein bestimmtes Gerät, auch wenn genau das gleiche Gerät auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

MOUSE_ID

BUTTONS

MOUSE_MODEL

MOUSE_TYPE

NATIV_SWARE

Beschreibung

Enthält Angaben zu einem bestimmten Betriebssystempaket; wird durch die Tabelle INST_NATIVE_SWARE ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für eine bestimmte Datei, auch wenn genau die gleiche Datei auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

NATIV_ID

PACKAGE_NAME

PACKAGE_VERS

PUBLISHER

PACKAGE_ID

NET_ADAPTER

Beschreibung

Enthält Angaben zu dem auf einem System installierten Netzwerkadapter. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

PERM_MAC_ADDR

CURRENT_ADDR

ADAPTER_TYPE

ADAPTER_MODEL

MANUFACTURER

INST_DATE

RECORD_TIME

NW_SERVER

Beschreibung

Enthält Angaben zu den Einstellungen auf einem NetWare-Server. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

NW_DEV_NAME

NW_VERS

NW_SUB_VERS

NW_MAX_CONNS

NW_MAX_VOLS

NW_REVISION_LEVEL

NW_SFT_LEVEL

NW_TTS_LEVEL

NW_MAX_CONNS_USED

NW_ACCOUNTING_VERS

NW_VAP_VERS

NW_QUEING_VERS

NW_PRINTSERVR_VERS

NW_VIRT_CONS

NW_SEC_LEVEL

NW_INET_BRG_SUPP

NW_CLIB_MAJOR_VERS

NW_CLIB_MINOR_VERS

NW_CLIB_REVISION

NW_SER_NUM

RECORD_TIME

NW_VOLS

Beschreibung

Enthält Angaben zu einem auf einem System installierten NetWare-Datenträger. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

NWVOL_NAME

NWVOL_TOTAL_BLKS

NWVOL_BLK_SECTORS

NWVOL_AVAIL_BLKS

NWVOL_DIR_SLOTS

NWVOL_AVAIL_SLOTS

NWVOL_IS_REMOVABLE

RECORD_TIME

PC_SYS_PARAMS

Beschreibung

Enthält die BIOS-Daten und andere Systemdaten eines PCs. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

USER_NAME

DOMAIN_NAME

WORKGROUP_NAME

BIOS_ID

BIOS_ID_BYTES

BIOS_DATE

BIOS_STRING

BIOS_MANUFACTURER

BIOS_MODEL

BIOS_SER_NUM

RECORD_TIME

PCI_DEV

Beschreibung

Enthält Angaben zu einem auf dem System installierten oder mit dem System verbundenen PCI-Gerät. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

INST_PCI_ID

PCI_DEV_NAME

RECORD_TIME

PRINTER

Beschreibung

Enthält Angaben zu einem bestimmten Druckertyp- und modell; wird durch die Tabelle INST_PRINTER ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für ein bestimmtes Gerät, auch wenn genau das gleiche Gerät auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

PRINTER_ID

PRINTER_MODEL

PROCESSOR

Beschreibung

Enthält Angaben zu einem bestimmten Prozessortyp- und modell; wird durch die Tabelle INST_PROCESSOR ergänzt. Das Konfigurations-Repository enthält nur eine Tabelle für

ein bestimmtes Gerät, auch wenn genau das gleiche Gerät auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

PROCESSOR_ID

MANUFACTURER

PROCESSOR_MODEL

PROCESSOR_FEATURES

MAX_SPEED

CURRENT_SPEED

CHIP_FAMILY

CHIP_MODEL

CHIP_STEPPING

VIRT_MODE_EXT

PAGE_SIZE_EXT

TIME_STAMP_COUNTER

MODEL_SPECIFIC_REG

PHYSICAL_ADDR_EXT

MACHINECHECK_EXCPT

CMPXCHG8B_SUPP

ON_CHIP_APIC

MEM_TYPE_RANGE_REG

PAGE_GLOBAL_ENABLE

MACHINECHECK_ARCH

COND_MOVE_SUPP

MMX_TECHNOLOGY

ON_CHIP_FPU

DEBUG_EXT_PRESENT

FAST_SYS_CALL

PAGE_ATTR_TABLE

PAGE_SIZE_EXT36

SER_NUM_ENABLED

FAST_FLOAT_SAVE

SIMD_EXT_SUPP

NOW_3_D_ARCH

STORAGE_DEV

Beschreibung

Enthält Angaben zu einem bestimmten auf dem System installierten Speichergerät; wird durch die Tabelle HDISK ergänzt. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

STORAGE_CLASS

INST_STORAGE_ID

STORAGE_TYPE

MANUFACTURER

MODEL

SER_NUM

HDISK_ID

RECORD_TIME

SWARE_SIG

Beschreibung

Enthält Angaben zu einer bestimmten Softwaresignatur; wird durch die Tabelle MATCHED_SWARE und die folgenden Ansichten ergänzt: H_INST_SWARE_VIEW, H_NOSIG_FILES_VIEW, INST_SWARE_VIEW, MATCH_SWARE-_VIEW, NOSIG_FILES_VIEW, SWARE_MATCH_CRC32, SWARE_MATCH_MD5 und SWARE_MATCH_QUICK. Das Konfigurations-Repository enthält nur eine Tabelle für eine bestimmte Datei, auch wenn genau die gleiche Datei auf mehreren Systemen in der TMR installiert ist.

Anmerkung

Wird durch den Befehl **winvsig** und über die GUI für Tivoli Inventory Version 4.0 ausgefüllt.

Spalten

SWARE_SIG_ID

SWARE_NAME

SWARE_SIZE

SWARE_DESC

SWARE_VERS

SIG_SOURCE

SIG_STATUS

CHECKSUM_QUICK

CHECKSUM_CRC32

CHECKSUM_MD5

RECORD_TIME

UNIX_SYS_PARAMS

Beschreibung

Enthält die UNIX-Systemparameter für ein System. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

BOOT_TIME

UPTIME

RUN_LEVEL

HOST_NAME

RECORD_TIME

UNMATCHED_FILES

Beschreibung

Enthält Angaben zu einer Datei, zu der keine Softwaresignatur passt; wird durch die Tabellen FILE_DESC und FILE_PATH ergänzt. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Softwaresuchlauf ausgefüllt.

Spalten

COMPUTER_SYS_ID

FILE_DESC_ID

INST_PATH_ID

CREATED_TIME

MODIFIED_TIME

ACCESSED_TIME

FILE_PERMISSIONS

FILE_OWNER

FILE_GROUP

CHECKSUM_QUICK

CHECKSUM_CRC32

CHECKSUM_MD5

RECORD_TIME

USB_DEV

Beschreibung

Enthält Angaben zu einem bestimmten USB-Gerätetyp- und modell; wird durch die Tabelle INST_USB_DEV ergänzt. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

USB_ID

USB_VERS

DEV_CLASS

DEV_SUBCLASS

VENDOR_ID

PRODUCT_ID

MANUFACTURER

PRODUCT

NUM_OF_PORTS

DEV_IS_HUB

VID_CARD

Beschreibung

Enthält Angaben zu einem bestimmten Videokartentyp- und modell; wird durch die Tabelle INST_VID_CARD ergänzt. Für jedes durchsuchte System ist eine Tabelle vorhanden.

Anmerkung

Wird durch einen Tivoli Inventory-Hardwaresuchlauf ausgefüllt.

Spalten

VID_CARD_ID

VID_CARD_MODEL

VID_CARD_BIOS

VID_DAC_TYPE

VID_MEM

VID_BIOS_RELDATE

VID_CHIP_TYPE

Protokolltabellen

Die Namen der Protokolltabellen bestehen aus ’H_’ und dem Namen der Betriebsdatentabelle, auf der die entsprechende Protokolltabelle basiert. Eine Protokolltabelle gibt alle Spalten zurück, die auch die entsprechende Betriebsdatentabelle zurückgibt, sowie zusätzlich die Spalten RECORD_ACTION und PRFL_ACTION.

In der folgenden Liste werden die verfügbaren Protokolltabellen aufgeführt:

H_COMPUTER

H_COMPUTER_SYS_MEM

H_INST_HEADER_INFO

H_INST_MOUSE

H_INST_NATIVE_SWARE

H_INST_PARTITION

H_INST_PRINTER

H_INST_PROCESSOR

H_INST_USB_DEV

H_INST_VID_CARD

H_IP_ADDR

H_IPX_ADDR

H_MATCHED_SWARE

H_NET_ADAPTER

H_NW_SERVER

H_NW_VOLS

H_PC_SYS_PARAMS

H_PCI_DEV

H_STORAGE_DEV

H_UNIX_SYS_PARAMS

H_UNMATCHED_FILES

D

Richtlinien

In diesem Anhang werden Richtlinien im Allgemeinen erläutert. Darüber hinaus wird hier auf die einzelnen Richtlinien eingegangen, die in Tivoli Inventory zur Anwendung kommen.

Bei Richtlinien handelt es sich um Regeln, die für die Ressourcen in Richtlinienbereichen maßgeblich sind. Zwei Arten von Richtlinien sind für die Regeln in einem Richtlinienbereich bestimmend:

- *Standardwertegruppe* - Sie ermöglicht die Festlegung der Standardwerte für verwaltete Ressourcen.
- *Richtlinie für Gültigkeitsprüfung* - Sie ermöglicht die Erstellung von Regeln, deren Einhaltung beim Hinzufügen bzw. Ändern verwalteter Ressourcen durch den Administrator erzwungen wird.

Weitere Informationen zu Richtlinien und Richtlinienbereichen können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen. Im vorliegenden Anhang finden Sie Hinweise zu den Standardwertegruppenmethoden, die in Tivoli Inventory zur Verfügung stehen, sowie ein Beispiel, das die Erstellung von Richtlinienmethoden und Richtlinienobjekten veranschaulicht.

Mit den im vorliegenden Abschnitt beschriebenen Standardwertegruppenmethoden in Tivoli Inventory werden die Standardwerte für die über **InventoryConfig** verwalteten Ressourcen festgelegt, d. h., über diese Methoden werden die Optionen festgelegt, die bei Erstel-

lung eines neuen Inventory-Profils standardmäßig übernommen werden. (Richtlinien für Gültigkeitsprüfungen stehen in Tivoli Inventory derzeit nicht zur Verfügung.)

Sollen die Eigenschaften von Inventory-Profilen auf bestimmte Werte gesetzt werden, können Sie die in Tivoli Inventory verfügbaren Standardwertegruppenmethoden bei Bedarf ändern. Wenn beispielsweise alle Inventory-Profile innerhalb eines bestimmten Richtlinienbereichs nur die seit dem letzten Suchlauf vorgenommenen Änderungen im Konfigurations-Repository übernehmen sollen, können Sie eine Richtlinienmethode definieren, die diese Option immer für neu erstellte Profile setzt.

Standardwertegruppenmethoden

Bei ***Standardwertegruppenmethoden*** handelt es sich um Shell-Skripts bzw. -Programme (UNIX-Skripts wie z. B. Bourne, K und Perl-Shell-Skripts, awk-Programme, C-Programme usw.), die bei Erstellung eines neuen Inventory-Profils von Tivoli Inventory aufgerufen werden. Indem Sie Skripts bzw. Programme erstellen und den Inhalt dieser Standardwertegruppenmethoden durch eigene Angaben ersetzen, können Sie festlegen, welche Optionen bei der Erstellung neuer Inventory-Profile automatisch gesetzt werden. Folgendes sollten Sie bei der Erstellung eigener Richtlinienmethoden berücksichtigen:

- Es wird empfohlen, Richtlinienprogramme in einer Interpreter-Sprache zu erstellen. Die Richtlinien werden in der Datenbank gespeichert, und Interpreter-Programme belegen in der Regeln weniger Speicherplatz als dies bei ausführbaren Dateien (kompilierten Programmen) der Fall ist. Darüber hinaus können Interpreter-Programmen plattformübergreifend eingesetzt werden, ausführbare Dateien hingegen nicht.
- Beim Aufrufen einer Standardwertegruppenmethode durch Tivoli Inventory wird der Name des zu erstellenden Inventory-Profils als erstes Argument an die Methode übergeben. Tivoli Inventory erwartet, dass die Standardwertegruppenmethode den Endecode **0** (exit 0) zurückgibt. Sie müssen die Richtlinienmethoden also

entsprechend erstellen. Für permanente Fehler wie beispielsweise Speicherengpässe, falsche Belegung usw. sollten daher andere Endecodes reserviert werden.

- C-Shell-Skripts sollten nicht verwendet werden, da Dateideskriptoren nicht in der für Tivoli Management Framework erforderlichen Weise behandelt werden.

Richtlinienobjekte

Bei einem Richtlinienobjekt handelt es sich um eine Gruppe von Richtlinienmethoden für eine bestimmte Ressourcenklasse. Die Standardwertegruppenmethoden, die für die Ressource **InventoryConfig** maßgeblich sind, werden in dem Standardrichtlinienobjekt **BasicInventoryConfig** definiert.

Das Richtlinienobjekt **BasicInventoryConfig** und die entsprechenden Richtlinienmethoden werden von Tivoli Inventory zur Verfügung gestellt. Darüber hinaus können Sie für die Ressource **InventoryConfig** weitere Richtlinienobjekte erstellen. Mehrere Richtlinienobjekte ermöglichen Ihnen die Definition verschiedener Richtlinien mit denselben Richtlinienmethoden.

Angenommen, Sie verfügen über zwei Richtlinienbereiche namens **Data** und **Software**. Für die Erstellung der Richtlinien, die die Standardwerte für Namen und Optionen von Inventory-Profilen in den beiden Richtlinienbereichen festlegen, können Sie beispielsweise die beiden Richtlinienobjekte **DataPolicy** und **SoftwarePolicy** erstellen. Nachdem Sie die Richtlinien für die Methoden in beiden Richtlinienobjekten festgelegt und die Richtlinienobjekte den entsprechenden Richtlinienbereichen zugeordnet haben, werden alle neuen Inventory-Profile entsprechend den von Ihnen definierten Richtlinien erstellt.

Gehen Sie bei der Definition eines neuen Richtlinienobjekts und dessen Richtlinienmethoden wie folgt vor:

- Erstellen Sie ein neues Richtlinienobjekt.
- Ersetzen Sie die Richtlinien für die Methoden dieses Richtlinienobjekts.

- Ordnen Sie das neue Richtlinienobjekt dem Richtlinienbereich zu, in dem sich die Inventory-Profile befinden werden.

In den folgenden Abschnitten finden Sie genaue Anweisungen zu den oben beschriebenen Schritten. Umfassende Informationen zum Überprüfen der Richtlinien in einem Richtlinienbereich können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen.

Neue Richtlinienobjekte erstellen

Voraussetzung für die Definition unterschiedlicher Richtlinien für mehrere Richtlinienbereiche ist die Erstellung neuer Richtlinienobjekte. Falls Sie für einen Richtlinienbereich keine Richtlinie definieren, verwendet Tivoli Inventory standardmäßig das Richtlinienobjekt **BasicInventoryConfig** und dessen Richtlinienmethoden.

In der folgenden Tabelle sind die zur Erstellung eines neuen Richtlinienobjekts erforderlichen Berechtigungsklassen aufgeführt:

Operation	Kontext	Erforderliche Berechtigungsklasse
Neues Richtlinienobjekt erstellen	TMR	**senior** oder **super**

Richtlinienobjekte müssen über die Befehlszeile erstellt werden.

Geben Sie zum Erstellen eines standardmäßigen Richtlinienobjekts den Befehl **wcrtpol** wie folgt ein:

```
wcrtpol -d InventoryConfig DataPolicy
```

Dabei gilt:

-d Gibt an, dass ein standardmäßiges Richtlinienobjekt erstellt werden soll.

InventoryConfig
Gibt **InventoryConfig** als Ressourcenart des neuen standardmäßigen Richtlinienobjekts an.

DataPolicy

Gibt **DataPolicy** als Name des neuen standardmäßigen Richtlinienobjekts an.

Neue Richtlinienobjekte erben die Richtlinienmethoden bereits vorhandener Richtlinienobjekte. Nach Erstellung eines neuen Richtlinienobjekts können Sie die vorhandenen Richtlinienmethoden anzeigen, um die Merkmale für Inventory-Profile bzw. die Operationen, die für einen bestimmten Richtlinienbereich ausgeführt werden, zu überprüfen. Verwenden Sie dazu folgende Befehle:

wlspolm

Gibt eine Liste der Richtlinienmethoden für die angegebene Ressource zurück. Mit diesem Befehl können Sie die Standardwertegruppenmethoden bzw. Richtlinien für die Gültigkeitsprüfung abrufen.

wgetpolm

Ruft den Inhalt der angegebenen Standardwertegruppenmethode bzw. Richtlinienmethode für die Gültigkeitsprüfung ab. Richtlinienmethoden werden von dem vorhandenen Richtlinienobjekt **InventoryConfig** geerbt.

Weitere Informationen zu den Befehlen **wcrtpol**, **wlspolm** und **wgetpolm** können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Inhalt von Richtlinienmethoden ersetzen

Sollen andere als die vom übergeordneten Richtlinienobjekt (dem Elternobjekt) geerbten Richtlinien definiert werden, müssen Sie ein Skript bzw. ein Programm erstellen, mit dem die bestehende Richtlinienmethode ersetzt wird.

In der folgenden Tabelle sind die Berechtigungsklassen aufgeführt, die erforderlich sind, um den Inhalt einer Richtlinienmethode zu ersetzen:

Operation	Kontext	Erforderliche Berechtigungs-klasse
Inhalt einer Richtlinien-methode ersetzen	TMR	**senior** oder **super**

Der Inhalt von Richtlinienobjekten muss über die Befehlszeile ersetzt werden.

Geben Sie den folgenden **wputpolm**-Befehl ein, um den Inhalt der Richtlinienmethode **ic_def_global_logfile_path** durch den Inhalt des Skripts **Data_logfile_def.sh** zu ersetzen:

```
wputpolm -d InventoryConfig DataPolicy \
ic_def_global_logfile_path < Data_logfile_def.sh
```

Dabei gilt:

-d Gibt an, dass es sich bei der Methode um eine Standardwertegruppenmethode handelt.

InventoryConfig
Gibt **InventoryConfig** als Ressourcenart an, für die die Richtlinie definiert ist.

DataPolicy
Gibt **DataPolicy** als Richtlinienobjekt an, das die standardmäßigen Richtlinienmethoden enthält, die ersetzt werden sollen.

ic_def_global_logfile_path
Ersetzt den Inhalt der standardmäßigen Richtlinienmethode **ic_def_global_logfile_path**.

< Data_logfile_def.sh
Leitet das Skript **Data_logfile_def.sh** an den Befehl um. Der Inhalt dieser Datei ersetzt den aktuellen Inhalt der Richtlinienmethode **ic_def_global_logfile_path**.

Weitere Informationen zum Befehl **wputpolm** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.

Richtlinie einem Richtlinienbereich zuweisen

Um die Standardwertegruppe eines Richtlinienbereichs zu ändern, müssen Sie nach Erstellung eines neuen Richtlinienobjekts und dem Ersetzen der Richtlinienmethoden die Richtlinie dem Richtlinienbereich zuweisen. In der folgenden Tabelle sind die Berechtigungsklassen aufgeführt, die zum Zuweisen einer Richtlinie zu einem Richtlinienbereich erforderlich sind:

Operation	Kontext	Erforderliche Berechtigungsklasse
Richtlinie einem Richtlinienbereich zuweisen	Richtlinienbereich	**senior** oder **super**

Sie können einem Richtlinienbereich Richtlinien über die Tivoli-Arbeitsoberfläche oder die Befehlszeile zuweisen. Anweisungen zum Arbeiten mit der Tivoli-Arbeitsoberfläche können Sie dem *Tivoli Management Framework Benutzerhandbuch* entnehmen. Um die Standardwertegruppe des Richtlinienbereichs **Dev** durch die im Richtlinienobjekt **DataPolicy** definierten Methoden zu ersetzen, müssen Sie den Befehl **wsetpr** wie folgt eingeben:

```
wsetpr -d DataPolicy InventoryConfig @PolicyRegion:Dev
```

Dabei gilt:

-d DataPolicy
: Gibt an, dass die Standardwertegruppe durch die im Objekt **DataPolicy** definierte Standardrichtlinie ersetzt werden soll.

InventoryConfig
: Gibt **InventoryConfig** als Ressourcenart an, für die die Richtlinie definiert ist.

@PolicyRegion:Dev
: Gibt **Dev** als den Richtlinienbereich an, dem die Richtlinie zugewiesen werden soll.

Weitere Informationen zum Befehl **wsetpr** finden Sie im Handbuch *Tivoli Management Framework Reference Manual*.

Standardwertegruppenmethode festlegen—Beispiel

Das folgende Beispiel zeigt, wie über die Befehlszeile die Standardrichtlinie für die Liste standardmäßiger Dateierweiterungen, nach denen gesucht werden soll, festgelegt sowie eine Richtlinie erstellt und einem neuen Richtlinienobjekt zugewiesen wird. Gehen Sie hierzu wie folgt vor:

1. Listen Sie die Standardwertegruppenmethoden für die Klasse **InventoryConfig** auf, indem Sie hinter der Root-Eingabeaufforderung den folgenden Befehl eingeben:

   ```
   wlspolm -d InventoryConfig
   ```

   Dabei gilt:

   -d Listet die Standardwertegruppenmethoden für die Ressourcenart **InventoryConfig** auf.

   InventoryConfig
   Gibt **InventoryConfig** als die Ressourcenart an, deren Richtlinienmethoden aufgelistet werden sollen.

   Die folgenden Standardwertegruppen werden zurückgegeben:

   ```
   ic_def_global_action
   ic_def_global_ep_timeout
   ic_def_global_logfile_host
   ic_def_global_logfile_path
   ic_def_global_notice_interval
   ic_def_global_notice_location
   ic_def_global_notice_type
   ic_def_global_update_replace
   ic_def_pc_custom_mifs
   ic_def_pc_custom_script
   ic_def_pc_exclude_dirs
   ic_def_pc_extensions
   ic_def_pc_hw_gran
   ic_def_pc_hw_outfile
   ic_def_pc_hw_scan
   ic_def_pc_include_dirs
   ic_def_pc_sw_crc
   ic_def_pc_sw_flags
   ic_def_pc_sw_outfile
   ic_def_pc_sw_scan
   ic_def_unix_custom_mifs
   ic_def_unix_exclude_dirs
   ic_def_unix_extensions
   ```

```
ic_def_unix_hw_gran
ic_def_unix_hw_outfile
ic_def_unix_hw_scan
ic_def_unix_include_dirs
ic_def_unix_sw_crc
ic_def_unix_sw_flags
ic_def_unix_sw_outfile
ic_def_unix_sw_scan
```

2. Listen Sie die Standardrichtlinienobjekte der Klasse **Inventory-Config** auf. Tivoli Management Framework unterstützt mehrere Standardrichtlinienobjekte, d. h. in den Richtlinienbereichen **X** und **Y** können jeweils unterschiedliche Standardrichtlinien zur Anwendung kommen. Mit dem folgenden Befehl können Sie die Standardrichtlinienobjekte für Tivoli Inventory auflisten:

   ```
   wlspol -d InventoryConfig
   ```

   Dabei gilt:

   -d Listet die Standardrichtlinienobjekte für die Ressourcenart **InventoryConfig** auf.

   InventoryConfig
   Gibt **InventoryConfig** als die Ressourcenart an, deren Richtlinienobjekte aufgelistet werden sollen.

   Haben Sie keine weiteren Standardrichtlinienobjekte erstellt, wird von diesem Befehl nur das Objekt **BasicInventoryConfig** zurückgegeben.

3. Extrahieren Sie den aktuellen Inhalt der Standardwertegruppenmethode **ic_def_pc_extensions** um sicherzustellen, dass sie von keinem Administrator geändert wurde.

   ```
   wgetpolm -d InventoryConfig BasicInventoryConfig \
   ic_def_pc_extensions
   ```

   Dabei gilt:

   -d Listet den Inhalt der Standardwertegruppenmethode auf.

InventoryConfig

Gibt **InventoryConfig** als die Ressource an, deren Richtlinie zurückgegeben werden soll.

BasicInventoryConfig

Gibt **BasicInventoryConfig** als das Richtlinienobjekt an, dessen Richtlinie zurückgegeben werden soll.

ic_def_pc_extensions

Gibt die Richtlinienmethode an, deren Inhalt zurückgegeben werden soll.

Der Inhalt dieser Richtlinienmethode wird standardmäßig an die Standardausgabeeinheit ausgegeben. Von diesem Befehl wird folgendes zurückgegeben:

```
#!/bin/sh
#  Component Name:

#  $Date: 2000/10/03 15:00:00 $
#
#  $Source: ic_def_pc_extensions,v $
#
#  $Id: ic_def_pc_extensions,v 1.0 2000/10/03 15:00:00 plw Exp $
#
#  Description:
#
#  Copyright 2000 by Tivoli Systems, Inc.  An IBM Company
#  Unpublished Work
#  All Rights Reserved
#  Licensed Material - Property of TIVOLI Systems, Inc.  An
#  IBM Company

EXCLUDE=0
INCLUDE=1

# MUST set Include or Exclude first before listing extensions.
# If no extensions are listed then just select on Include or Exclude.
# It will not matter.
echo $INCLUDE

# PC extensions MUST be in uppercase.  No validation will be done.
echo "*.EXE
*.DLL
*.COM
*.NLM"

exit 0
```

4. Erstellen Sie ein Skript, das die Erweiterungsliste in eine EXCLUDE-Liste ändert und die Liste mit den Erweiterungen ignoriert. Dies wird vom Skript /tmp/list.sh ausgeführt:

```
#!/bin/sh
# Custom policy

EXCLUDE=0
INCLUDE=1

# MUST set Include or Exclude first before listing extensions.
# If no extensions are listed then just select on Include or Exclude.
# It will not matter.
echo $EXCLUDE

echo "*.BAK
*.TMP
*.SYS
*.INI"

exit 0
```

5. Ersetzen Sie mit dem folgenden Befehl den Inhalt der Richtlinienmethode **ic_def_pc_extensions** durch das neue Skript:

```
wputpolm -d InventoryConfig BasicInventoryConfig \
ic_def_pc_extensions </tmp/list.sh
```

Dabei gilt:

-d Gibt an, dass es sich bei der Richtlinienmethode **ic_def-_pc_extensions** um eine Standardwertegruppenmethode handelt.

InventoryConfig
Gibt **InventoryConfig** als die Ressource an, für die die Richtlinie festgelegt wird.

BasicInventoryConfig
Gibt **BasicInventoryConfig** als das Richtlinienobjekt an, für das die Richtlinie festgelegt wird.

ic_def_pc_extensions
Gibt **ic_def_pc_extensions** als die Richtlinienmethode an, deren Inhalt ersetzt werden soll.

</tmp/list.sh

Leitet das Skript /tmp/list.sh an den Befehl um. Die Eingabe für diesen Befehl wird aus der Standardeingabeeinheit gelesen.

6. Ordnen Sie die neue Richtlinienmethode im Richtlinienobjekt **BasicInventoryConfig** dem Richtlinienbereich **Dev** wie folgt zu:

```
wsetpr -d BasicInventoryConfig InventoryConfig \
@PolicyRegion:Dev
```

Dabei gilt:

-d BasicInventoryConfig

Gibt an, dass die Standardwertegruppe durch die im Objekt **BasicInventoryConfig** definierte Standardrichtlinie ersetzt werden soll.

InventoryConfig

Gibt **InventoryConfig** als Ressourcenart an, für die die Richtlinie definiert ist.

@PolicyRegion:Dev

Gibt **Dev** als den Richtlinienbereich an, dem die Richtlinie zugeordnet werden soll.

Nach dem Zuordnen der Richtlinie wird in allen Inventory-Profilen, die in den Richtlinienbereichen erstellt werden, als deren Standardwertegruppe für die Ressourcenart **InventoryConfig** das Richtlinienobjekt **BasicInventoryConfig**angegeben wurde, die Liste mit den im Beispiel-Skript angeführten Dateierweiterungen übernommen.

Richtlinienmethoden

Mit den folgenden Standardwertegruppenmethoden können Sie die Standardeinstellungen der über **InventoryConfig** verwalteten Ressourcen steuern. Diese Methoden können von Bereich zu Bereich verschieden sein, d. h. in den einzelnen Richtlinienbereichen kommen jeweils andere Standardwertegruppenmethoden zur Anwendung.

Anmerkung: Die Standardwertegruppenmethoden für Abfragen sind hier ebenfalls aufgeführt, obwohl sie zum Lieferumfang von Tivoli Management Framework gehören.

Bei erfolgreicher Ausführung wird als Endecode von den Richtlinienmethoden der Wert Null, anderenfalls ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_action

Gibt an, ob nur die Konfigurationsdatei verteilt werden soll oder ob sie verteilt und ein Suchlauf ausgeführt werden soll.

ZUSAMMENFASSUNG

ic_def_global_action

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt an, ob ein Inventory-Profil nur die Konfigurationsdatei verteilt oder ob sie verteilt und auch ein Suchlauf ausgeführt wird. Wenn Sie einen unterbrochenen Suchlauf ausführen, möchten Sie u.U. nur die Konfigurationsdatei verteilen.

Folgende Werte sind für diese Methode zulässig:

16 Gibt an, dass nur die Konfigurationsdatei verteilt wird.

17 Gibt an, dass die Konfigurationsdatei verteilt und ein Suchlauf ausgeführt wird. Dies ist der Standardwert.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_ep_timeout

Gibt die standardmäßige Zeitspanne an, nach der ein Suchlauf auf einem Endpunkt automatisch beendet werden soll.

ZUSAMMENFASSUNG

ic_def_global_ep_timeout

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird der Standardwert des Attributs endpt_time-_out festgelegt. Tivoli Inventory legt die Zeitdauer eines Suchlaufs auf jedem Endpunkt fest. Wenn der Suchlauf vor Ablauf der von dieser Methode festgelegten Zeitdauer nicht abgeschlossen wird (der Endpunkt gibt keinen Status für den Suchlauf zurück), wird der Suchlauf auf diesem Endpunkt beendet, und Tivoli Inventory zeigt an, dass der Suchlauf auf diesem Endpunkt nicht erfolgreich ausgeführt wurde.

Die für diese Methode zulässigen Werte sind Dezimalzahlen größer als Null. Der Standardwert beträgt 1800 Sekunden.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_logfile_host

Gibt den Namen des verwalteten Knotens an, auf dem die Protokolldatei mit den Tivoli Inventory-Statusinformationen gespeichert werden soll.

ZUSAMMENFASSUNG

ic_def_global_logfile_host

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt den Namen des verwalteten Knotens an, auf dem die Protokolldatei mit den Tivoli Inventory-Statusinformationen geschrieben werden soll. Sie wird nur verwendet, wenn ein Inventory-Profil so konfiguriert wurde, dass die Statusinformationen in eine Protokolldatei geschrieben werden. Wird kein verwalteter Knoten angegeben, wird als Standardoption der verwaltete Knoten verwendet, auf dem der Inventory Data Handler installiert ist. Wenn ein Name angegeben wird, muss er mit dem Namen eines gültigen verwalteten Knotens übereinstimmen. Beispiel:

```
echo "rainier"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_logfile_path

Gibt den Pfad und Namen der Protokolldatei mit den Tivoli Inventory-Statusinformationen an.

ZUSAMMENFASSUNG

ic_def_global_logfile_path

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt den Pfad und Namen der Protokolldatei mit den Tivoli Inventory-Statusinformationen an. Sie wird nur verwendet, wenn ein Inventory-Profil so konfiguriert wurde, dass die Statusinformationen in eine Protokolldatei geschrieben werden. Wenn diese Methode keinen Wert angibt, erhält die Datei den Namen inv_scan_*n*, wobei *n* für die ID-Nummer des Suchlaufs steht. Die Datei wird im temporären Verzeichnis des verwalteten Knotens gespeichert, der in ic_def_global_logfile_host angegeben ist. Dieses temporäre Verzeichnis hat in der Regel einen der folgenden Pfade:

- Für UNIX: /tmp, /usr/tmp oder /var/tmp
- Für Windows: c:\temp

Beispiel:

```
 echo "/tmp/pc_scan_hw.log"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_notice_interval

Gibt an, wann eine Benachrichtigung für einen Suchlauf mit Statusbericht gesendet wird.

ZUSAMMENFASSUNG

ic_def_global_notice_interval

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt an, wann für einen Suchlauf eine Benachrichtigung gesendet werden soll. Optional können Sie angeben, ob eine Benachrichtigung bei jedem abgeschlossenen Ziel, zu bestimmten Zeitintervallen oder bei allen abgeschlossenen Zielen gesendet werden soll.

Folgende Werte sind für diese Methode zulässig:

1 Benachrichtigung bei jedem abgeschlossenen Ziel senden.

16 Benachrichtigung nach einer bestimmten Zeitdauer senden und/oder Benachrichtigung senden, nachdem eine bestimmte Anzahl von Zielen abgeschlossen wurde. Die Zeitdauer und die Anzahl der abgeschlossenen Ziele richtet sich nach den Einstellungen des Inventory Data Handler.

256 Benachrichtigung erst nach Abschluss aller Ziele senden. Dies ist der Standardwert.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Richtlinienmethoden geben bei Fehlern einen Endecode mit einem Wert ungleich Null zurück. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_notice_location

Gibt die Adressen an, an die die Benachrichtigungen gesendet werden.

ZUSAMMENFASSUNG

ic_def_global_notice_location

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode zeigt an, wohin die Statusinformationen für ein Profil während eines Suchlaufs geschrieben werden sollen. Sie können eine oder mehrere der folgenden Optionen angeben:

Folgende Werte sind für diese Methode zulässig:

1 Statusinformationen in die Inventory-Hinweisgruppe schreiben. Dies ist der Standardwert.

16 Statusinformationen in eine Protokolldatei schreiben.

256 Statusinformationen in Tivoli Enterprise Console (TEC) schreiben.

Kein Wert bzw. der Wert Null zeigt an, dass keine Statusinformationen geschrieben werden sollen. Sie können die Informationen auch an mehrere Adressen senden, indem Sie die entsprechenden Werte zum Schreiben der Statusinformationen addieren. Im nachfolgenden Beispiel werden die Statusinformationen in die Inventory-Hinweisgruppe und in TEC geschrieben:

```
echo 257
oderexpr 256 + 1
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_notice_type

Gibt die Art der Zielsuchläufe an, für die eine Benachrichtigung gesendet werden soll.

ZUSAMMENFASSUNG

ic_def_global_notice_type

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt einen numerischen Wert zurück, der die Umstände anzeigt, unter denen eine Benachrichtigung generiert wird.

Folgende Werte sind für diese Methode zulässig:

1 Benachrichtigung für erfolgreiche Ziele senden.

16 Benachrichtigung für fehlgeschlagene Ziele senden.

Sie können auch Benachrichtigungen sowohl für erfolgreiche als auch für fehlerhafte Ziele senden, indem Sie den Wert 17 (16 + 1) angeben. Wenn Sie den Wert 0 angeben, werden keine Benachrichtigungen gesendet. Der Wert 17 ist der Standardwert.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_global_update_replace

Gibt an, ob Tivoli Inventory alle Daten oder nur die Änderungen ersetzen soll, wenn das Konfigurations-Repository mit den Ergebnissen des Suchlaufs aktualisiert wird.

ZUSAMMENFASSUNG

ic_def_global_update_replace

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird festgelegt, ob Tivoli Inventory während eines Suchlaufs alle Daten im Konfigurations-Repository ersetzt oder die aktuellen Suchlaufdaten mit der MIF-Datei vergleicht, die bei dem vorherigen Suchlauf generiert wurde, und nur die Änderungen ersetzt. Wenn ein Endpunkt noch über keine Suchlaufdaten in der Datenbank verfügt, werden alle Daten für diesen Endpunkt an das Konfigurations-Repository gesendet.

Folgende Werte sind für diese Methode zulässig:

1 Gibt an, dass das Ergebnis des aktuellen Suchlaufs mit dem des letzten Suchlaufs verglichen wird und nur die Abweichungen im Konfigurations-Repository gesichert werden. Dies ist die Standardoption.

16 Gibt an, dass das Ergebnis des aktuellen Suchlaufs im Konfigurations-Repository gesichert werden soll.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_custom_mifs

Gibt den Pfad und den Namen der MIF-Benutzerdateien an, die während des Suchlaufs auf einem PC gelesen werden sollen.

ZUSAMMENFASSUNG

ip_def_custom_mif_file

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert die Liste der MIF-Benutzerdateien, die Tivoli Inventory von den PC-Zielrechnern einliest. Standardmäßig werden keine MIF-Benutzerdateien gelesen. Werden in dieser Methode mehrere MIF-Benutzerdateien angegeben, muss jeder Dateiname in einer neuen Zeile aufgeführt werden. Verzeichnisnamen müssen, wie im nachfolgenden Beispiel gezeigt, zwischen Schrägstrichen (/) stehen:

```
echo "useradd.mif
C:/scandata/data.mif"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_custom_script

Gibt die Standardskripts an, die während des PC-Suchlaufs ausgeführt werden sollen.

ZUSAMMENFASSUNG

ic_def_pc_custom_script

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode gibt die Benutzerskripts an, die während des PC-Suchlaufs ausgeführt werden sollen. Die Skripts werden nach Beendigung der Tivoli-Suchsoftware, jedoch vor dem Lesen der MIF-Dateien ausgeführt. In der Regel gibt diese Methode BAT- oder CMD-Skripts an. Beispiel:

```
echo "C:/MYSCAN.EXE
COPY C:/MYSCAN.MIF D:/TIVOLI/USERADD.MIF
```

Standardmäßig werden keine Benutzerskripts ausgeführt.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_exclude_dirs

Gibt die Liste der Verzeichnisse an, die standardmäßig bei einem PC-Softwaresuchlauf ausgeschlossen werden.

ZUSAMMENFASSUNG

ic_def_pc_exclude_dirs

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird die Liste mit den Verzeichnissen generiert, die bei einem PC-Softwaresuchlauf standardmäßig nicht von einem Inventory-Profil durchsucht werden. Die Methode schreibt eine Liste der Verzeichnisnamen in die Standardausgabeeinheit, wobei jeder Name in einer neuen Zeile aufgeführt wird. Standardmäßig werden die Verzeichnisse */TMP, */TEMP, /RECYCLED/ und /RECYCLER/ nicht durchsucht.

Verzeichnisnamen müssen in Großbuchstaben geschrieben werden und zwischen Schrägstrichen stehen. Beispiel:

```
echo "*/CACHE
*/BACKUP"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_extensions

Gibt eine Liste mit Dateien oder Dateitypen an, die während eines PC-Softwaresuchlaufs durchsucht bzw. ignoriert werden sollen.

ZUSAMMENFASSUNG

ic_def_pc_extensions

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode stellt eine Liste mit Dateinamen (z. B. file.exe) und/oder Dateitypen (z. B. *.exe) bereit. Sie generiert einen Wert, der angibt, ob die Dateien oder Dateitypen während eines PC-Softwaresuchlaufs durchsucht oder ignoriert werden sollen.

Standardmäßig werden die Dateitypen *.exe, *.dll, *.com, *.nlm und *.sig bei PC-Softwaresuchläufen durchsucht.

Die erste Zeile der Methodenausgabe enthält eine der folgenden Optionen:

```
0     Liste mit Erweiterungen bei einem PC-Softwaresuchlauf ignorieren
1     Liste mit Erweiterungen bei einem PC-Softwaresuchlauf durchsuchen
```

Die nachfolgenden Ausgabezeilen enthalten die Liste mit Dateinamen oder -typen, die jeweils in einer neuen Zeile stehen. Siehe dazu folgendes Beispiel:

```
echo 1
echo "*.EXE
*.DLL
*.DRV
*.NLM
CONFIG.SYS"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_hw_gran

Gibt die Hardwarekomponenten an, nach denen ein PC durchsucht wird.

ZUSAMMENFASSUNG

ic_def_pc_hw_gran

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert eine Liste der Hardwarekomponenten, nach denen während eines PC-Suchlaufs gesucht werden kann, und weist jeder Komponente einen Wert zu. Der Wert zeigt an, ob Informationen zu dieser Komponente gesammelt werden sollen. Nachfolgend wird der Inhalt der von Tivoli verwendeten Methode aufgeführt. Die Liste enthält alle Komponentennamen, die Sie in Ihrer Methode verwenden können.

```
$/bin/sh
OFF=0
ON=1
echo "Processor $ON"
echo "Memory $ON"
echo "Operating System $ON"
echo "Storage $ON"
echo "IP Address $ON"
echo "Network Adapter $ON"
echo "Partition $ON"
echo "Pointing Device $ON"
echo "Keyboard $ON"
echo "IPX Address $ON"
echo "Video $ON"
echo "Printer $ON"
echo "USB Device $ON"
exit 0
```

Standardmäßig werden alle Hardwarekomponenten aus dieser Liste durchsucht.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_hw_outfile

Gibt die Tivoli MIF-Dateien an, die während eines Hardwaresuchlaufs von einem PC abgerufen werden.

ZUSAMMENFASSUNG

ic_def_pc_hw_outfile

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche MIF-Dateien und somit welche Suchlaufdaten von einem PC-Endpunkt abgerufen und während eines Hardwaresuchlaufs an das Konfigurations-Repository geendet werden. Folgende Werte sind für diese Methode zulässig:

0 Keine MIF-Dateien während des Hardwaresuchlaufs zurückgeben.

1 Die von der Tivoli-Suchsoftware erstellten Hardware-MIF-Dateien zurückgeben. Dies ist die Standardoption.

16 Die von der DMI-Suchsoftware erstellten Hardware-MIF-Dateien zurückgeben.

Sie können auch festlegen, dass beide MIF-Dateien zurückgegeben werden sollen, indem Sie die entsprechenden Werte für den Tivoli-Hardwaresuchlauf und den DMI-Suchlauf addieren. Beispiel:

```
echo 17
oderexpr 16 + 1
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_hw_scan

Gibt an, welche Hardwaresuchläufe beim Durchsuchen der PC-Endpunkte ausgeführt werden sollen.

ZUSAMMENFASSUNG

ic_def_pc_hw_scan

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche Hardwaresuchläufe während des PC-Suchlaufs ausgeführt werden sollen. Folgende Werte sind für diese Methode zulässig:

0 Keinen Hardwaresuchlauf ausführen.

1 Tivoli-Suchsoftware ausführen. Dies ist die Standardoption.

16 DMI-Suchsoftware ausführen.

Sie können die Tivoli- und die DMI-Suchläufe ausführen, indem Sie deren jeweilige Werte addieren. Beispiel:

```
echo 17
oderexpr 16 + 1
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_include_dirs

Gibt die Liste der Verzeichnisse an, die standardmäßig bei einem PC-Softwaresuchlauf durchsucht werden.

ZUSAMMENFASSUNG

ic_def_pc_include_dirs

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird die Liste mit den Verzeichnissen generiert, die bei einem PC-Softwaresuchlauf standardmäßig durchsucht werden. Die Methode schreibt eine Liste der Verzeichnisnamen in die Standardausgabeeinheit, wobei jeder Name in einer neuen Zeile aufgeführt wird. PC-Verzeichnisnamen müssen in Großbuchstaben geschrieben werden und zwischen Schrägstrichen (/) stehen. Beispiel:

```
echo "*/PROGRAMME
*/APPLICATIONS"
exit 0.
```

Standardmäßig werden keine Verzeichnisse angegeben (alle Verzeichnisse werden durchsucht).

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_sw_crc

Gibt an, welcher zyklische Blockprüfungswert (Cyclic Redundancy Check, CRC), falls vorhanden, während eines PC-Softwaresuchlaufs für Dateien ermittelt wird.

ZUSAMMENFASSUNG

ic_def_pc_sw_crc

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welcher CRC-Wert während eines PC-Softwaresuchlaufs für Dateien ermittelt wird. Während eines Suchlaufs kann nur ein CRC-Wertetyp ermittelt werden. Gültig sind folgende Werte:

0 Keinen CRC-Wert für Dateien berechnen. Dies ist die Standardoption.

1 4-Byte-Kontrollsumme für die ersten 1024 Bytes der Datei erstellen.

16 4-Byte-Kontrollsumme für alle Bytes der Datei erstellen.

256 16-Byte-MD5-Kontrollsumme für eine Datei erstellen.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_sw_flags

Gibt an, welche Sonderoptionen während des PC-Softwaresuchlaufs aktiviert sind.

ZUSAMMENFASSUNG

ic_def_pc_sw_flags

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche Sonderoptionen während eines PC-Softwaresuchlaufs aktiviert sind. Die verfügbaren Optionen zeigen an, ob nur nach ausführbaren Dateien gesucht wird und ob ein Benutzerfilter auf die grundlegenden Informationen der Softwaresuche angewendet werden soll. Folgende Optionswerte stehen zur Verfügung:

0 Keine Sonderoptionen beim PC-Softwaresuchlauf. Dies ist die Standardoption.

1 Nur nach ausführbaren Dateien suchen.

16 Benutzerfilter auf die grundlegenden Informationen des Softwaresuchlaufs anwenden.

Sie können auch mehrere Optionen angeben, indem Sie die Summe der gewünschten Optionen generieren. Mit folgendem Wert suchen Sie beispielsweise nur nach ausführbaren Dateien und wenden einen Benutzerfilter an:

```
echo 17
oderexpr 1 + 16
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_sw_outfile

Gibt die Tivoli-MIF-Dateien an, die während eines Softwaresuchlaufs von PCs abgerufen werden.

ZUSAMMENFASSUNG

ic_def_pc_sw_outfile

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche MIF-Dateien und somit welche Suchlaufdaten von einem PC-Endpunkt abgerufen und während eines Softwaresuchlaufs an das Konfigurations-Repository geendet werden. Folgende Werte sind für diese Methode zulässig:

0 Keine MIF-Dateien zurückgeben.

1 Die MIF-Datei mit den grundlegenden Dateiinformationen zurückgeben.

16 Die MIF-Datei mit den Datei-Header-Informationen zurückgeben.

256 Die MIF-Datei mit den Registrierungsdatenbankinformationen zurückgeben.

4096 Die Software-MIF-Datei zurückgeben, die durch Abgleich der Softwaresignatur erstellt wurde.

Sie können auch festlegen, dass mehrere MIF-Dateien zurückgegeben werden sollen, indem Sie die entsprechenden Werte der gewünschten MIF-Dateien für den Tivoli-Hardwaresuchlauf addieren. Mit folgendem Wert werden beispielsweise die grundlegenden Dateiinformationen, die Datei-Header-Informationen und die PC-Registrierungsdatenbankinformationen zurückgegeben:

```
echo 273
oderexpr 256 + 16 + 1
```

Der Standardwert ist 272 (256 + 16).

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_pc_sw_scan

Gibt an, welche Softwaresuchläufe beim Durchsuchen der PC-Endpunkte ausgeführt werden sollen.

ZUSAMMENFASSUNG

ic_def_pc_sw_scan

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche Softwaresuchläufe während des PC-Softwaresuchlaufs ausgeführt werden sollen. Folgende Werte sind für diese Methode zulässig:

0 Keine Softwaresuchläufe ausführen.

1 Nach grundlegenden Dateiinformationen suchen.

16 Nach Datei-Header-Informationen suchen.

256 Die PC-Registrierungsdatenbank durchsuchen. Dies ist die Standardoption.

4096 Nach Dateisignaturinformationen suchen.

Sie können auch mehrere Suchläufe ausführen, indem Sie die entsprechenden Werte für die gewünschten Suchläufe addieren. Mit folgendem Wert führen Sie beispielsweise Suchläufe in der PC-Registrierungsdatenbank, nach Datei-Header-Informationen und nach grundlegenden Dateiinformationen aus:

```
echo 273
oder
expr 256 + 16 + 1
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_custom_mifs

Gibt den Pfad und den Namen der MIF-Benutzerdateien an, die während des Suchlaufs auf UNIX-Systemen gelesen werden.

ZUSAMMENFASSUNG

ic_def_unix_custom_mifs

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert die Liste der MIF-Benutzerdateien, die Tivoli Inventory von den UNIX-Zielrechnern abruft. Standardmäßig werden keine MIF-Benutzerdateien abgerufen. Wenn Sie mehrere MIF-Benutzerdateien auflisten, müssen die Dateinamen in der Ausgabe jeweils in einer neuen Zeile stehen. Beispiel:

```
echo "useradd.mif
/usr/apps/custom.mif"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_custom_script

Gibt die Benutzerskripts an, die während des Suchlaufs auf einem UNIX-System ausgeführt werden sollen.

ZUSAMMENFASSUNG

ic_def_unix_custom_script

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert ein Skript, das während eines Suchlaufs ausgeführt werden soll (nach Abschluss des Suchlaufs, jedoch vor Lesen der MIF-Dateien). Bei UNIX-Systemen handelt es sich in der Regel um ein .sh-Skript. Beispiel:

```
echo "#!/bin/sh
/usr/local/bin/myscan.sh
cp /tmp/scandata.mif /tivoli/useradd.mif
"
```

Standardmäßig werden keine Skripts ausgeführt.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_exclude_dirs

Gibt die Liste der Verzeichnisse an, die standardmäßig bei einem Softwaresuchlauf auf UNIX-Systemen ausgeschlossen werden.

ZUSAMMENFASSUNG

ic_def_unix_exclude_dirs

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird die Liste mit den Verzeichnissen generiert, die bei einer Profilverteilung an ein UNIX-System standardmäßig nicht von einem Inventory-Profil durchsucht werden. Standardmäßig wird das Verzeichnis */tmp nicht durchsucht.

Die Methode schreibt eine Liste der Verzeichnisnamen in die Standardausgabeeinheit, wobei jeder Name in einer neuen Zeile aufgeführt wird. Beispiel:

```
echo "*/tmp
*/temp/"
exit 0
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_extensions

Gibt eine Liste mit Dateinamen oder -typen an, die während eines Softwaresuchlaufs auf UNIX-Systemen durchsucht bzw. ignoriert werden sollen.

ZUSAMMENFASSUNG

ic_def_unix_extensions

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode stellt eine Liste mit Dateinamen (z. B. readme.txt) und/oder Dateitypen (z. B. *.txt) bereit. Sie generiert einen Wert, der angibt, ob die Dateinamen oder -typen während des Softwaresuchlaufs auf einem UNIX-System durchsucht oder ignoriert werden sollen. Die erste Zeile der Methodenausgabe enthält eine der folgenden Optionen:

```
0     Liste mit Erweiterungen bei einem UNIX-Softwaresuchlauf ignorieren
1     Liste mit Erweiterungen bei einem UNIX-Softwaresuchlauf durchsuchen
```

Die nachfolgenden Ausgabezeilen enthalten die Liste mit Dateinamen oder -typen, die jeweils in einer neuen Zeile stehen. Siehe dazu folgendes Beispiel:

```
echo 1
echo "*.txt
myfile
*.TXT"
exit 0
```

Standardmäßig werden keine Dateinamen oder -typen angegeben (alle Dateien werden während des Suchlaufs durchsucht).

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_hw_gran

Gibt die Hardwarekomponenten an, nach denen ein UNIX-System durchsucht wird.

ZUSAMMENFASSUNG

ic_def_unix_hw_gran

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert eine Liste der Hardwarekomponenten, nach denen während eines UNIX-Suchlaufs gesucht werden kann, und weist jeder Komponente einen Wert zu. Der Wert zeigt an, ob Informationen zu dieser Komponente gesammelt werden sollen. Nachfolgend wird der Inhalt der von Tivoli verwendeten Methode aufgeführt. Die Liste enthält alle Komponentennamen, die Sie in Ihrer Methode verwenden können.

```
#!/bin/sh
OFF=0
ON=1
echo "Processor $ON"
echo "Memory $ON"
echo "Operating System $ON"
echo "Storage $ON"
echo "IP Address $ON"
echo "Network Adapter $ON"
echo "Partition $ON"
echo "Pointing Device $ON"
echo "Keyboard $ON"
echo "Unix System Params $ON"
exit 0
```

Standardmäßig werden alle Hardwarekomponenten aus dieser Liste durchsucht.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_hw_outfile

Gibt die Tivoli-MIF-Dateien an, die während eines Hardwaresuchlaufs von einem UNIX-System abgerufen werden.

ZUSAMMENFASSUNG

ic_def_unix_hw_outfile

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, ob die MIF-Datei mit den Hardwaresuchlaufdaten von einem UNIX-Endpunkt abgerufen und während eines Hardwaresuchlaufs an das Konfigurations-Repository geendet wird. Folgende Werte sind für diese Methode zulässig:

0 Keine Hardware-MIF-Datei während eines Hardwaresuchlaufs zurückgeben.

1 Die Hardware-MIF-Datei zurückgeben. Dies ist die Standardoption.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_hw_scan

Gibt an, ob ein Hardwaresuchlauf auf UNIX-Endpunkten durchgeführt werden soll.

ZUSAMMENFASSUNG

ic_def_unix_hw_scan

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, ob ein Hardwaresuchlauf auf UNIX-Endpunkten durchgeführt werden soll. Gültig sind folgende Werte:

0 Keinen Hardwaresuchlauf auf UNIX-Endpunkten ausführen.

1 Hardwaresuchlauf auf UNIX-Endpunkten ausführen. Dies ist die Standardoption.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_include_dirs

Gibt die Liste der Verzeichnisse an, die standardmäßig bei einem Softwaresuchlauf auf UNIX-Systemen durchsucht werden.

ZUSAMMENFASSUNG

ic_def_unix_include_dirs

RESSOURCE

InventoryConfig

BESCHREIBUNG

Über diese Methode wird die Liste mit den Verzeichnissen generiert, die bei einem Softwaresuchlauf auf UNIX-Systemen standardmäßig durchsucht werden. Die Methode schreibt eine Liste der Verzeichnisnamen in die Standardausgabeeinheit, wobei jeder Name in einer neuen Zeile aufgeführt wird. Die Verzeichnisnamen müssen zwischen Schrägstrichen (/) stehen. Beispiel:

```
echo "/usr/local/bin
*/bin"
exit 0
```

Standardmäßig werden keine Verzeichnisse angegeben (alle Verzeichnisse werden durchsucht).

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_sw_crc

Gibt an, welcher zyklische Blockprüfungswert (Cyclic Redundancy Check, CRC), falls vorhanden, während eines Softwaresuchlaufs auf UNIX-Systemen für Dateien ermittelt wird.

ZUSAMMENFASSUNG

ic_def_unix_sw_crc

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welcher CRC-Wert während eines UNIX-Softwaresuchlaufs für Dateien ermittelt wird. Während eines Suchlaufs kann nur ein CRC-Wertetyp ermittelt werden. Gültig sind folgende Werte:

0 Keinen CRC-Wert für Dateien berechnen. Dies ist die Standardoption.

1 4-Byte-Kontrollsumme für die ersten 1024 Bytes der Datei erstellen.

16 4-Byte-Kontrollsumme für alle Bytes der Datei erstellen.

256 16-Byte-MD5-Kontrollsumme für eine Datei erstellen.

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_sw_flags

Gibt an, welche Sonderoptionen während des Softwaresuchlaufs auf UNIX-Systemen aktiviert sind.

ZUSAMMENFASSUNG

ic_def_unix_sw_flags

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche Sonderoptionen während eines Softwaresuchlaufs auf UNIX-Systemen aktiviert sind. Die verfügbaren Optionen zeigen an, ob nur nach ausführbaren Dateien gesucht wird und ob ein Benutzerfilter auf die grundlegenden Informationen des Softwaresuchlaufs angewendet werden soll. Folgende Optionswerte stehen zur Verfügung:

0 Keine Sonderoptionen beim UNIX-Softwaresuchlauf.

1 Nur nach ausführbaren Dateien suchen. Dies ist die Standardoption.

16 Benutzerfilter auf die grundlegenden Informationen des Softwaresuchlaufs anwenden.

Sie können auch mehrere Optionen angeben, indem Sie die Summe der gewünschten Optionen generieren. Mit folgendem Wert suchen Sie beispielsweise nur nach ausführbaren Dateien und wenden einen Benutzerfilter an:

```
echo 17
oderexpr 1 + 16
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_sw_outfile

Gibt die Tivoli-MIF-Dateien an, die während eines Softwaresuchlaufs von einem UNIX-System abgerufen werden.

ZUSAMMENFASSUNG

ic_def_unix_sw_outfile

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche MIF-Dateien und somit welche Suchlaufdaten von einem UNIX-Endpunkt abgerufen und während eines Softwaresuchlaufs an das Konfigurations-Repository gesendet werden. Folgende Werte sind für diese Methode zulässig:

0 Keine MIF-Dateien zurückgeben.

1 Die MIF-Datei mit den grundlegenden Dateiinformationen zurückgeben.

256 Die MIF-Datei mit den Informationen zum installierten Produkt und zu Programmkorrekturen zurückgeben. Dies ist die Standardoption.

4096 Die Software-MIF-Datei zurückgeben, die durch Abgleich der Softwaresignatur erstellt wurde.

Sie können auch festlegen, dass mehrere MIF-Dateien zurückgegeben werden sollen, indem Sie die entsprechenden Werte der gewünschten MIF-Dateien addieren. Mit folgendem Wert werden beispielsweise die grundlegenden Dateiinformationen und die Softwaresignaturinformationen zurückgegeben:

```
echo 4097
oderexpr 4096 + 1
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

ic_def_unix_sw_scan

Gibt an, welche Softwaresuchläufe beim Durchsuchen von UNIX-Endpunkten ausgeführt werden sollen.

ZUSAMMENFASSUNG

ic_def_unix_sw_scan

RESSOURCE

InventoryConfig

BESCHREIBUNG

Diese Methode generiert einen Dezimalwert, der angibt, welche Softwaresuchläufe während eines UNIX-Softwaresuchlaufs ausgeführt werden sollen. Folgende Werte sind für diese Methode zulässig:

0 Keine Softwaresuchläufe ausführen.

1 Nach grundlegenden Dateiinformationen suchen.

256 Nach Informationen zum installierten Produkt und zu Programmkorrekturen suchen. Dies ist die Standardoption.

4096 Nach Dateisignaturinformationen suchen.

Sie können auch mehrere Suchläufe ausführen, indem Sie die entsprechenden Werte für die gewünschten Suchläufe addieren. Mit folgendem Wert suchen Sie beispielsweise nach den grundlegenden Dateiinformationen und nach Informationen zum installierten Produkt und zu Programmkorrekturen:

```
echo 257
oderexpr 256 + 1
```

Bei erfolgreicher Ausführung geben die Tivoli Inventory-Richtlinienmethoden den Endecode **0** zurück. Anderenfalls wird ein Wert ungleich Null zurückgegeben. Die von der Methode zurückgegebenen Informationen müssen als Zeichenfolge in die Standardausgabeeinheit geschrieben werden.

E

Fehlerbehebung

In diesem Anhang werden Dateien und Prozeduren beschrieben, die Ihnen bei der Untersuchung und Behebung von Problemen, die möglicherweise bei der Anwendung von Tivoli Inventory auftreten, helfen können. Grundsätzlich sollten Sie zunächst die in der Tivoli Inventory-Hinweisgruppe zur Fehlerbehebung enthaltenen Informationen beachten, bevor Sie die in diesem Anhang beschriebenen möglichen Fehlerquellen überprüfen. Informationen zu Hinweisgruppen finden Sie im *Tivoli Management Framework Benutzerhandbuch*.

Konfigurations-Repository-Schema

Um Fehler zu beheben, die während der Erstellung des Konfigurations-Repositorys auftreten, überprüfen Sie die Protokolldatei mit dem Namen inv_*RDBMS-Typ*_admin.log, wobei *RDBMS-Typ* für den von Ihnen für das Konfigurations-Repository verwendeten RDBMS-Typ steht. Diese Variable kann einer der folgenden Werte sein: db2, ms, ora oder syb. In der Protokolldatei wird angegeben, ob die Ausführung der SQL-Anweisungen erfolgreich verlaufen ist oder nicht. Überprüfen Sie Ihre RDBMS-Dokumentation, um die Adresse dieser Protokolldatei festzustellen.

DMI

Dieser Abschnitt enthält Informationen zur Fehlerbehebung bei DMI-Suchläufen.

DMI-Ebene hinzufügen

Bevor Sie nach DMI-Daten suchen können, müssen Sie zunächst DMI-Suchfunktionen konfigurieren. Hierfür müssen Sie im Dialog 'Konfiguration für DMI-Suchfunktion' alle DMI-Ebenen, die Sie durchsuchen möchten, hinzufügen. Stellen Sie vor dem Versuch, eine DMI-Ebene hinzuzufügen, sicher, dass Sie ein Inventory-Profil mindestens einmal an den Endpunkt, auf dem die DMI-Ebene installiert ist, verteilt haben. Ein Inventory-Profil verteilt folgende Dateien an einen Endpunkt: dmiscan.exe, wcdmi.dll und wdmiutil.dll. Diese Dateien müssen auf dem Endpunkt vorhanden sein, bevor Sie die DMI-Ebene abrufen können.

Bei dem Versuch, eine DMI-Ebene im Dialog 'Konfiguration für DMI-Suchfunktion' hinzuzufügen, wird möglicherweise einer der Fehler aus der nachstehenden Liste angezeigt. Zu jeder Fehlernachricht werden Informationen zur Fehlerbehebung gegeben.

- **Endpunkt konnte nicht gefunden werden**.

  Geben Sie zur Behebung dieses Fehlers folgenden Befehl ein, um zu überprüfen, ob der Endpunkt verfügbar ist:

  ```
  wadminep Endpunktname
  view_log_file
  ```

  Dabei gilt:

  Endpunktname
  : Gibt den Namen des Endpunkts an.

  view_log_file
  : Zeigt eine Protokolldatei für den Endpunkt an.

  Kann die Protokolldatei nicht angezeigt werden, ist der Endpunkt nicht verfügbar. Überprüfen Sie den Endpunkt sowie die Verbindung zwischen dem Endpunkt und dem Gateway, dem er zugeordnet ist.

- **DMI-Ebene konnte nicht untersucht werden**.

  Eine DMI-Ebene ist auf dem angegebenen Endpunkt vorhanden, jedoch ist das für die Untersuchung der DMI-Ebene zulässige Zeitlimit überschritten worden. Wiederholen Sie die Operation zu einem späteren Zeitpunkt.

- **Auf diesem Endpunkt ist keine DMI-Ebene vorhanden**.

  Für den von Ihnen angegebenen Endpunkt ist keine DMI-Ebene vorhanden. Installieren Sie eine DMI-Ebene auf diesem Endpunkt, oder geben Sie einen anderen Endpunkt an, der demselben Typ der DMI-Ebene angehört.

- **DMI-Daten konnten nicht abgerufen werden**.

  Es ist ein Problem in der Kommunikation zwischen dem Endpunkt und dem Gateway, dem er zugeordnet ist, aufgetreten. Geben Sie folgenden Befehl ein, um diesen Fehler zu beheben:

  ```
  wadminep Endpunktname view_log_file
  ```

  Dabei gilt:

  Endpunktname
  Gibt den Namen des Endpunkts an.

  view_log_file
  Zeigt eine Protokolldatei für den Endpunkt an.

  Kann die Protokolldatei nicht angezeigt werden, ist der Endpunkt nicht verfügbar. Überprüfen Sie den Endpunkt sowie die Verbindung zwischen dem Endpunkt und dem Gateway, dem er zugeordnet ist.

- **DMI-Datenformat wird nicht erkannt.**

  Führen Sie folgende Task aus, um diesen Fehler zu beheben:

  - Geben Sie einen anderen Endpunkt an, der demselben Typ der DMI-Ebene angehört.
  - Untersuchen Sie die DMI-Ebene auf dem Endpunkt mit einem DMI-Browser, wie beispielsweise dem Intel DMI Explorer, um festzustellen, ob die Daten beschädigt sind oder einen nicht erkannten Wert enthalten.

- Installieren Sie die DMI-Ebene auf dem Endpunkt erneut.

Weitere Informationen zur Verwendung des Dialogs 'Konfiguration für DMI-Suchfunktion' und zur Konfiguration der DMI-Suchlaufoptionen finden Sie in der Onlinehilfe von Tivoli Inventory.

DMI-Suchlauf

Schlägt ein DMI-Suchlauf fehl, führen Sie die folgende Task aus, um den Fehler zu beheben:

- Prüfen Sie, ob der Endpunkt, auf dem der Suchlauf fehlgeschlagen ist, über eine DMI-Ebene verfügt. Bei PCs, die nicht über eine DMI-Ebene verfügen, gibt ein DMI-Suchlauf den Wert `NO_DATA` für alle Attribute zurück. Um zu überprüfen, ob eine DMI-Ebene auf einem Endpunkt installiert ist, führen Sie die Prozedur zum Hinzufügen der DMI-Ebene vom Endpunkt zu der Liste der DMI-Ebenen im Dialog 'Konfiguration für DMI-Suchfunktion' aus. Ist auf dem Endpunkt keine DMI-Ebene installiert, wird die folgende Nachricht angezeigt:

  ```
  Auf diesem Endpunkt ist keine DMI-Ebene vorhanden.
  ```

  Weitere Informationen zur Prozedur zum Hinzufügen einer DMI-Ebene finden Sie in der Onlinehilfe zu Tivoli Inventory.

- Stellen Sie sicher, dass die Datei dmi.ini an den Endpunkt verteilt wurde. Diese Datei enthält die Einstellungen zur DMI-Suchfunktion. Das Inventory-Profil muss diese Datei erfolgreich an den Endpunkt verteilen, damit ein DMI-Suchlauf beendet werden kann.

- Führen Sie den DMI-Suchlauf auf dem Endpunkt über die Befehlszeile aus. Die Adresse für die DMI-Suchfunktion lautet $LCF_INSTDIR/inv/SCAN, wobei $LCF_INSTDIR das Verzeichnis ist, auf dem der Endpunkt installiert ist. Führen Sie den DMI-Suchlauf mit folgendem Befehl aus:

  ```
  dmiscan -i dmi.ini -o dmiscan.mif
  ```

DMI-Tabellen- und Spaltennamen

Das Skript dmi_*RDBMS-Typ*_schema.sql erstellt im Konfigurations-Repository benutzerdefinierte Tabellen für Informationen zu DMI-Suchläufen. Bei dem Standardnamen für jede benutzerdefinierte Tabelle handelt es sich um den auf 18 Zeichen gekürzten DMI-Gruppennamen; Leerstellen werden dabei durch Unterstriche ersetzt. Bei den Standardnamen für Spalten in jeder Tabelle handelt es sich um die auf 18 Zeichen gekürzten DMI-Attributnamen; Leerstellen werden dabei durch Unterstriche ersetzt.

Da diese Namen automatisch generiert werden, ist es möglich, dass das Skript als Tabellen- und Spaltennamen reservierte Wörter von SQL-92 verwendet. In diesem Fall müssen Sie die Tabelle oder Spalte umbenennen. Weitere Informationen zu reservierten Wörtern von SQL-92 erhalten Sie in der SQL-Dokumentation oder bei einem Datenbankadministrator.

Ebenfalls ist es möglich, dass das Skript ungültige Zeichen in Tabellen- oder Spaltennamen verwendet. Namen für Tabellen und Spalten dürfen aus den folgenden Zeichen bestehen:

- a bis z in Kleinschreibung
- A bis Z in Großschreibung
- Numerische Zeichen von 0 bis 9
- Unterstreichung

Es dürfen keine Leerzeichen verwendet werden. Tabellen- und Spaltennamen müssen mit einem Buchstaben beginnen.

Endpunkt-initiierte Suchläufe

Um Fehler bei einem Endpunkt-initiierten Suchlauf zu beheben, ist es hilfreich, die Unterschiede zu einem durch Profilverteilung eingeleiteten Suchlauf zu kennen. Ein durch Profilverteilung eingeleiteter Suchlauf besteht aus den folgenden Schritten:

- Der Endpunkt wird durchsucht; die Suchlaufdaten werden auf dem Endpunkt in einer MIF-Datei gespeichert. Der Name der MIF-Datei lautet *Suchlauftyp*.mif, wobei *Suchlauftyp* den Typ des Suchlaufs bezeichnet. So generiert beispielsweise die Tivoli-Hardwaresuchfunktion die Datei tivhscan.mif, und die DMI-Suchfunktion generiert die Datei dmiscan.mif.
- Wenn das Inventory-Profil so konfiguriert ist, dass es die Ergebnisse des Suchlaufs an das Konfigurations-Repository sendet, wird die MIF-Datei mit Hilfe der Endpunktmethode syntaktisch analysiert und durch die Erweiterung BK1 umbenannt. Die auf diese Weise aus der MIF-Datei gewonnenen Daten werden in eine DAT-Datei (eine codierte und komprimierte Binärdatei) geschrieben. Der Name der DAT-Datei lautet inv*scanID*.dat, wobei *scanID* für den numerischen Wert steht, mit dem der Suchlauf identifiziert wird.
- Die .DAT-Datei wird über die einzelnen Ebenen der Empfangshierarchie an das Konfigurations-Repository gesendet.

Die Suchläufe, die von einem Endpunkt eingeleitet werden, werden über den Befehl **wepscan** gesteuert. Bei diesem Suchlauftyp wird die MIF-Datei erst dann durch die Erweiterung BK1 umbenannt, nachdem die Daten mit dem Befehl **wepscan** und unter Angabe der Option **–s** an das Konfigurations-Repository gesendet wurden. Anders ausgedrückt bedeutet dies, dass bei der Durchführung zweier aufeinander folgender Endpunkt-initiierter Suchläufe die während des ersten Suchlaufs generierte MIF-Datei durch den zweiten Suchlauf überschrieben wird, wenn die Daten nicht zuvor an das Konfigurations-Repository gesendet wurden. Die Daten aus dem ersten Suchlauf werden niemals erfasst. Dies soll gewährleisten, dass die BK1-Datei immer die jeweils zuletzt an das Konfigurations-Repository gesendeten Daten (dies sind nicht notwendigerweise die Daten aus dem letzten Suchlauf) enthält. Durch diese Vorgehensweise ist sichergestellt, dass der Suchlauf bei Auswahl der Option ’Nur Änderungen aktualisieren’ die Datenbank immer mit den korrekten Informationen aktualisiert. (Ein Suchlauf mit der Option ’Nur Änderungen aktualisieren’ vergleicht die MIF-Datei des laufenden Suchlaufs mit dem BK1-Suchlauf und sendet ausschließlich neue oder geänderte Daten an das Konfigurations-Repository.)

Bei Ausführung des Befehls **wepscan** mit der Option **–d** werden Protokolldateien mit den Namen sa_results.log und sa_config.log erstellt. Bei Ausführung des Befehls **wepscan** mit der Option **–c** wird eine Protokolldatei mit dem Namen sa_config.log erstellt. Diese Protokolldateien werden in dem Verzeichnis gespeichert, von dem aus der **wepscan**-Befehl ausgeführt wird. Die Protokolldateien enthalten folgende Informationen:

sa_results.log
: Den Inhalt der DAT-Datei. Prüfen Sie in dieser Datei, welche Daten durch den Suchlauf erfasst wurden. Die Daten werden an das Konfigurations-Repository gesendet, sobald die Suchlaufdaten erfasst sind.

sa_config.log
: Die Konfiguration des Profils. Prüfen Sie in dieser Datei, welche Software- und Hardwareinformationen das Profil auf Grund seiner Konfiguration erfasst und welche weiteren Profilkonfigurationsdaten enthalten sind.

Stellen Sie vor der Ausführung eines Endpunkt-initiierten Suchlaufs sicher, dass folgende Tasks durchgeführt wurden:

- Es wurde ein Inventory-Profil an den Endpunkt verteilt. Hierdurch wird die Konfigurationsdatei config.dmp auf dem Endpunkt installiert. Diese Datei muss vorhanden sein, damit der Endpunkt-initiierte Suchlauf erfolgreich durchgeführt werden kann. Sehen Sie in dem Verzeichnis, in dem der Endpunkt installiert ist, und dort im Verzeichnis inv/SCAN nach, um zu prüfen, ob die Datei config.dmp auf dem Endpunkt vorhanden ist.
- Ihre Umgebung wurde so konfiguriert, dass der für diesen Befehl erforderliche Zugriff auf eine DLL-Datei möglich ist. Weitere Informationen zur Konfiguration Ihrer Umgebung für die Verwendung von **wepscan** finden Sie unter „wepscan” auf Seite 271.

Grafische Benutzerschnittstelle

Dieser Abschnitt enthält Anweisungen zur Fehlerbehebung bei der grafischen Benutzerschnittstelle (GUI) von Tivoli Inventory.

Architekturvoraussetzungen

Für die Verwendung der grafischen Benutzerschnittstelle (GUI) von Tivoli Inventory Version 4.0 ist Java Runtime Environment (JRE) 1.2.2 erforderlich, das von der Tivoli Management Framework Produkt-CD installiert werden kann. Nicht alle von Tivoli Inventory unterstützten Betriebssysteme sind mit JRE 1.2.2. kompatibel. Lesen Sie die *Tivoli Inventory Release Notes, Version 4.0* und die Dokumentation für Ihr Betriebssystem, um weitere Informationen zu erhalten.

Die Protokolldatei aktivieren

Sie können die grafische Benutzerschnittstelle (GUI) von Tivoli Inventory so konfigurieren, dass Nachrichten in eine Protokolldatei geschrieben werden. Anschließend können Sie mit Hilfe der Protokolldatei Fehler in der grafischen Benutzerschnittstelle beheben. Diese Protokolldatei heißt DEBUG3 und ist im Verzeichnis $DBDIR gespeichert. Um die Protokolldatei für die grafische Benutzerschnittstelle zu aktivieren, ist folgende Prozedur erforderlich:

1. Öffnen Sie die Datei inv_gui.sh in einem Texteditor auf dem System, auf welchem Sie die grafische Benutzerschnittstelle verwenden. Die Adresse dieser Datei lautet: $BINDIR/TME/INVENTORY.
2. Suchen Sie am Ende der Datei inv_gui.sh eine Textstelle ähnlich dem folgenden Beispiel:

```
if [ "$INTERP" = "w32-ix86" ]
then
#  use for debugging
#  eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND" >>
#  $DBDIR/DEBUG3 2>&1
eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND"
else
#   eval "$COMMAND" >> $DBDIR/DEBUG3 2>&1
eval "$COMMAND"
fi
```

3. Löschen Sie das Kommentarzeichen aus der Zeile, die den Befehl für die Erstellung der Protokolldatei DEBUG3 enthält und fügen Sie ein Kommentarzeichen in die Zeile ein, die diesen Befehl nicht enthält.

   - Suchen Sie folgende Zeilen bei Verwendung von Windows-Plattformen:

     ```
     #  eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND" >>
     #  $DBDIR/DEBUG3 2>&1
     eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND"
     ```

     Editieren Sie die Zeilen wie folgt:

     ```
     eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND" >>
     # $DBDIR/DEBUG3 2>&1
     # eval "$INSTALL_DIR/bin/wrunui.exe $COMMAND"
     ```

   - Suchen Sie bei Verwendung von UNIX-Plattformen folgende Zeilen:

     ```
     #   eval "$COMMAND" >> $DBDIR/DEBUG3 2>&1
     eval "$COMMAND"
     ```

     Editieren Sie die Zeilen wie folgt:

     ```
     eval "$COMMAND" >> $DBDIR/DEBUG3 2>&1
     # eval "$COMMAND"
     ```

Die grafische Benutzerschnittstelle (GUI) aktivieren

Um die grafische Benutzerschnittstelle (GUI) von Tivoli Inventory auf einem UNIX-System auszuführen, müssen Sie Verbindungen zum X-Window-System auf dem UNIX-System aktivieren. Dies ist auch dann erforderlich, wenn die Konsole auf derselben Maschine wie der X-Window-Bildschirm ausgeführt wird. Gehen Sie hierzu wie folgt vor:

1. Legen Sie für die Umgebungsvariable DISPLAY den X-Window-Bildschirm fest, auf dem die Konsole ″Verteilungsstatus″ angezeigt werden soll.

   Um beispielsweise die Konsole auf dem X-Window-Bildschirm mit dem Namen zeus:0.0 zu öffnen, gibt ein Bourne-Shell- oder Korn-Shell-Benutzer folgende Befehle ein:

   ```
   DISPLAY=zeus:0.0
   export DISPLAY
   ```

Manchmal kann es erforderlich sein, als Variable DISPLAY eine IP-Adresse anstelle des Systemnamens anzugeben.

2. Aktivieren Sie Fernverbindungen zum X-Window-System. Um beispielsweise die Konsole ″Verteilungsstatus″ auf dem Bildschirm zeus zu starten, geben Sie folgenden Befehl ein:

   ```
   xhost +zeus
   ```

Stellen Sie sicher, dass die Funktion für Fernanmeldungen auf dem System aktiviert ist, auf dem Sie die Konsole starten. Geben Sie hierzu den folgenden Befehl ein:

```
odadmin odinfo
```

Daraufhin wird die folgende Nachricht angezeigt:

```
Remote client login allowed = Wert
```

wobei *Wert* entweder **TRUE** oder **FALSE** ist.

Wird der Wert **FALSE** angezeigt, geben Sie den folgenden Befehl ein, um die Funktion für Client-Fernanmeldungen zu aktivieren:

```
odadmin set_allow_rconnect TRUE
```

Scalable Collection Service (SCS)

Dieser Abschnitt enthält Informationen zur Fehlerbehebung beim Scalable Collection Service (SCS).

Protokoll- und Datendateien

In den folgenden Abschnitten werden Protokoll- und Datendateien im Zusammenhang mit SCS beschrieben. Sofern kein anderes Verzeichnis angegeben ist, befinden sich diese Dateien in einem der folgenden Verzeichnisse:

- Im Inventory Data Handler - ($DBDIR/inventory/mc/*Dateiname*)
- In allen weiteren Kollektoren - ($DBDIR/mcollect/*Dateiname*)

Dabei gilt:

$DBDIR

Ist das Tivoli-Objektdatenbankverzeichnis auf dem verwalteten Knoten

Dateiname

Steht für einen der in den folgenden Abschnitten aufgelisteten Dateinamen

Datendateien von Warteschlangen

Der SCS protokolliert Informationen zu Warteschlangendaten eines Kollektors in den folgenden Warteschlangen-Datendateien:

- checkpointGL_iqfile.dat - Listet die CTOCs (Collection Tables of Contents) in der Eingabewarteschlange des Kollektors auf.
- checkpointGL_oqfile.dat - Listet die CTOCs (Collection Tables of Contents) in der Ausgabewarteschlange des Kollektors auf.
- checkpointGL_eqfile.dat - Listet die CTOCs (Collection Tables of Contents) in der Fehlerwarteschlange des Kollektors auf. CTOCs werden in die Fehlerwarteschlange eines Kollektors verschoben, wenn der Versuch eines Kollektors, Informationen zu verschicken oder zu erhalten, fehlschlug. Die CTOC bleibt so lange in der Fehlerwarteschlange, bis der Kollektor die maximale Anzahl von Wiederholungen ausgeschöpft hat. Erst dann sendet der Kollektor einen Fehlerhinweis und entfernt die CTOC aus allen seinen Warteschlangen.
- checkpointGL_dqfile.dat - Listet die CTOCs (Collection Tables of Contents) in der verzögerten Warteschlange des Kollektors auf. CTOCs sind in der verzögerten Warteschlange vorhanden, wenn Verbindungen zum Kollektor getrennt sind. Weitere Informationen zum Trennen von Verbindungen finden Sie in den Abschnitten „Datenerfassung terminieren” auf Seite 109 und „wcollect” auf Seite 247.

Anmerkung: Die DAT-Dateien für die Warteschlangen sind vorhanden, so dass SCS beim Abschalten eines Kollektors wiederhergestellt werden kann und die Verarbeitung der erfassten Daten fortsetzt.

Mit dem Befehl **wcstat –q** können Sie diese Dateien anzeigen.

Diese Dateien haben eine Ausgangsgröße von 4 Bytes, die zunimmt, wenn die zugeordnete Warteschlange Daten enthält. Sie können die Größe dieser Dateien zur Fehlerbehebung bei Warteschlangen überprüfen. Um beispielsweise festzustellen, ob die Eingabewarteschlange Daten enthält, können Sie prüfen, ob die Warteschlangen-Datendatei checkpointGL_iqfile.dat größer als 4 Bytes ist.

Protokolldatei des Kollektors

Die Datei mcollect.log enthält alle Vorgänge, die ein Kollektor bei der Datenübertragung ausführt. Sie können die Anzahl der Daten, die in dieser Datei protokolliert werden, steuern, indem Sie einen Wert über die Option **wcollect –d** festlegen. Standardmäßig werden nur schwer wiegende Fehler protokolliert. Sie können die maximale Größe dieser Datei über die Option**–g** des Befehls **wcollect** festlegen. Der Standardwert ist 1 MB. Weitere Informationen können Sie dem Abschnitt „wcollect” auf Seite 247 entnehmen.

Sie können die Protokollvorgänge überwachen, indem Sie auf dem Kollektor den Befehl **tail** ausführen:

```
tail -f mcollect.log
```

Daraufhin zeigt die Protokolldatei eine Liste der Protokollvorgänge an. Wird die Anzeige der Protokolldatei während der Datenerfassung unterbrochen, hat die Datei ihre maximale Größe erreicht. Geben Sie in diesem Fall den Befehl **tail** erneut ein, um die Überwachung der Protokollvorgänge fortzusetzen.

Achten Sie während der Überwachung der Protokollvorgänge in der Datei mcollect.log auf die Zeichenfolgen WARNING, ERROR und ’exception’, die auf Fehlerbedingungen hinweisen.

Protokolldatei des Inventory Data Handler

Der Inventory Data Handler erstellt eine Protokolldatei, die mit der von jedem Kollektor erstellten Protokolldatei vergleichbar ist. Der Dateiname ist ebenfalls mcollect.log und Konfiguration und Überwachung dieser Protokolldatei erfolgen über die gleichen Befehle wie für die Kollektor-Protokolldatei.

Achten Sie während der Überwachung dieser Protokolldatei auf die Zeichenfolge IR, die auf eine Nachricht eines Inventory Data Handler hinweist. Weitere Informationen zur Konfiguration und Überwachung dieser Protokolldatei finden Sie im Abschnitt „Protokolldatei des Kollektors" auf Seite 522.

Gibt es Probleme bei der Verbindung des Inventory Data Handler zum Konfigurations-Repository, können Sie in der Datei mcollect.log prüfen, ob Nachrichten eingegangen sind. Der Inventory Data Handler schreibt in diese Protokolldatei Nachrichten, in denen angegeben ist, mit welchem RIM-Objekt er eine Verbindung zum Konfigurations-Repository herzustellen versucht.

Protokolldatei des Kollektor-Managers

Der Kollektor-Manager ist auf demselben System installiert wie der TMR-Server (Tivoli Management Region, TMR). Die Protokolldatei des Kollektor-Managers ist die Datei $TMPDIR/mcollect_collmgr.log auf dem TMR-Server, wobei $TMPDIR ein temporäres Verzeichnis ist. Dieses temporäre Verzeichnis hat in der Regel einen der folgenden Pfade:

- Für UNIX: /tmp, /usr/tmp oder /var/tmp
- Für Windows NT: c:\temp

Diese Protokolldatei enthält die Route-Informationen, die der Kollektor-Manager an die Kollektoren weiterleitet. Mit dem Befehl **tail** oder einem Texteditor können Sie die Datei anzeigen. Wenn Sie die Datei mit dem Befehl **tail** anzeigen möchten, führen Sie den Befehl folgendermaßen aus:

```
tail -f mcollect_collmgr.log
```

Tivoli Inventory-Statusprotokolldatei

Standardmäßig sendet Tivoli Inventory Statusinformationen über Suchläufe nur an die Tivoli Inventory-Hinweisgruppe. Mit dem Befehl **wsetinvglobal –l** können Sie Tivoli Inventory dahingehend konfigurieren, dass Statusinformationen an eine Protokolldatei gesendet werden.

Für diese Datei ist der Pfad $TMPDIR/inv_scan_*nn*.log als Standardeinstellung vorgegeben, wobei *nn* für die ID des Suchlaufs und $TMPDIR für ein temporäres Verzeichnis steht. Das temporäre Verzeichnis hat in der Regel einen der folgenden Pfade:

- Für UNIX: /tmp, /usr/tmp oder /var/tmp
- Für Windows NT: c:\temp

Anmerkung: Auf UNIX-Systemen ist $TMPDIR das temporäre Verzeichnis, das vom Systemaufruf **tmpnam** zurückgegeben wird. Auf Windows NT-Systemen ist $TMPDIR das temporäre Verzeichnis, das vom Systemaufruf **GetTempPath** zurückgegeben wird. Weitere Informationen finden Sie in der Dokumentation für Ihr Betriebssystem.

Diese Protokolldatei gibt den Profilnamen, die Startzeit, die abgelaufene Zeit und die Anzahl der Ziele an, auf denen Suchläufe beendet sind oder anstehen.

Mit dem Befehl **tail** oder einem Texteditor können Sie die Datei anzeigen. Wenn Sie die Datei mit dem Befehl **tail** anzeigen möchten, führen Sie den Befehl folgendermaßen aus:

```
tail -f inv_scan_nn.log
```

Statusprotokolldatei der beendeten CTOCs

Sie können einen Kollektor dahingehend konfigurieren, dass er eine Protokolldatei verwaltet, welche die auf diesem Kollektor beendeten CTOCs auflistet. Zur Fehlerbehebung können Sie in dieser Datei überprüfen, ob eine Datenerfassung auf einem Kollektor erfolgreich beendet wurde.

Der Pfad für diese Datei ist $DBDIR/mcollect/CTOC_log.dat. Geben Sie den folgenden Befehl ein, um den Inhalt dieser Datei anzuzeigen:

```
wcstat -q c Kollektor
```

Dabei hat *Kollektor* das folgende Format: **@ManagedNode:***Kollektor* bzw. **@Gateway:***Kollektor*.

Mit der Option **wcollect –f** können Sie angeben, ob der Kollektor Daten in die Datei **CTOC_log.dat** schreiben soll. Standardmäßig schreiben Kollektoren Daten in diese Datei. Um zu prüfen, ob ein Kollektor dahingehend konfiguriert ist, dass er Daten in diese Datei schreibt, führen Sie den folgenden Befehl aus:

```
wcollect Kollektor
```

Dabei hat *Kollektor* das folgende Format: **@ManagedNode:***Kollektor* bzw. **@Gateway:***Kollektor*.

Prüfen Sie im Abfrageergebnis den Wert von log_completed_ctoc. Der Wert TRUE gibt an, dass der Kollektor Daten in die Protokolldatei schreibt, und der Wert FALSE gibt an, dass keine Daten in die Datei geschrieben werden.

Weitere Protokolldateien

Die folgenden Protokolldateien enthalten Informationen, wie beispielsweise Fehlernachrichten, die bei der Fehlerbehebung hilfreich sein können. Diese Protokolldateien sind Textdateien, die Sie mit dem Befehl **tail** oder einem Texteditor anzeigen können.

- $LCF_DATDIR/lcfd.log - Eine Protokolldatei für Tivoli Management Framework-Nachrichten zur Aufzeichnung von Endpunktnachrichten. Diese Datei ist auf jedem Endpunkt vorhanden. Zur Fehlerbehebung beim SCS können Sie in dieser Datei prüfen, ob im Zusammenhang mit SCS Ausnahmebedingungen oder Probleme hinsichtlich der Konnektivität vorliegen. Das Verzeichnis $LCF_DATDIR wird bei der Installation von Tivoli Management Framework erstellt. Weitere Informationen zur Protokolldatei lcfd.log können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

- /tmp/rim_db_log- Die Protokolldatei mit Trace-Informationen für RDBMS Interface Module-Objekte (RDBMS Interface Module = RIM). Diese Datei befindet sich auf dem verwalteten Knoten, auf dem das RIM-Objekt installiert ist. Mit dem Befehl **wrimtrace** können Sie die Trace-Funktion aktivieren oder inaktivieren sowie die Informationen in der Protokolldatei angeben. In die RIM-Protokolldatei können Sie IOM-Paketinformationen (IOM, Interobject Message), Fehler im Zusammenhang mit Verwaltungssystemen für relationale Datenbanken (Relational Database Management System, RDBMS) oder beides schreiben.

  Die RIM-Protokolldatei wird nicht standardmäßig erstellt. Damit die RIM-Protokolldatei erstellt wird, müssen Sie mit dem Befehl **wrimtrace** die Trace-Funktion aktivieren.

  Anmerkung: Die Trace-Funktion ist für Debug-Vorgänge vorgesehen. Ist diese Funktion über einen längeren Zeitraum aktiviert, kann dies den Durchsatz von RIM-Aufrufen mindern und ihre Verarbeitung erheblich verlangsamen.

  Weitere Informationen zum Befehl **wrimtrace** und zur RIM-Protokolldatei können Sie dem Handbuch *Tivoli Management Framework Reference Manual* entnehmen.

Depot-Inhalt

Sie können den Inhalt eines Kollektor-Depots im Verzeichnis depot/*CTOC_ID* prüfen, wobei *CTOC_ID* für die Identifikationsnummer eines CTOC steht. Das Verzeichnis, das die Daten eines CTOCs enthält, hat als Namen die ID des betreffenden CTOCs. Als Standardeinstellung ist für das Depotverzeichnis der Pfad $DBDIR/mcollect/depot auf jedem Kollektor angegeben.

Die im Depotverzeichnis enthaltenen Dateien können nicht angezeigt werden. Sie können jedoch prüfen, ob das Verzeichnis vorhanden ist und ob es Daten enthält. Dies können Sie auch anhand der Dateigröße feststellen, die größer als 0 Byte ist, wenn in den Dateien Daten enthalten sind.

Verfahren zur Fehlerbehebung

Dieser Abschnitt bietet Lösungsmöglichkeiten bei Problemen im Zusammenhang mit der Verwendung von SCS. Weitere Informationen zu den in diesem Abschnitt beschriebenen Befehlen finden Sie unter „Befehle” auf Seite 239 und im Handbuch *Tivoli Management Framework Reference Manual*.

- **Der Suchlauf ist beendet, jedoch sind keine Daten in der Datenbank enthalten.**

  In einer SCS-Umgebung wird ein Inventory-Profil beendet, nachdem alle Ziele zwar physisch durchsucht wurden, die Daten jedoch nicht notwendigerweise an das Konfigurations-Repository weitergegeben wurden.

  Mit dem Befehl **wgetscanstat** wird ein Suchlauf nur dann als beendet angegeben, wenn alle erfassten Daten auch im Konfigurations-Repository gespeichert wurden oder wenn ein Fehler aufgetreten ist. Um festzustellen, ob die Daten für einen Endpunkt an das Konfigurations-Repository weitergegeben wurden, müssen Sie den Befehl folgendermaßen ausführen:

  ```
  wgetscanstat -a -s -f
  ```

Sie können auch in der Tivoli Inventory-Hinweisgruppe oder im Tivoli Inventory-Statusprotokoll prüfen, ob ein Suchlauf beendet ist, sofern das Profil für den Suchlauf so konfiguriert ist, dass Meldungen an diese Quellen weitergegeben werden.

- **Der Suchlauf sollte längst beendet sein.**

  Zeigen Sie zunächst mit dem Befehl **wgetscanstat** den Fortschritt des Suchlaufs auf den verschiedenen Endpunkten an. Wenn für die meisten Endpunkte der Suchlauf beendet ist, prüfen Sie mit dem Befehl **wadminep**, ob ein Zugriff auf die Endpunkte möglich ist, auf denen der Suchlauf noch nicht beendet ist.

  Prüfen Sie anschließend mit dem Befehl **wrimtest**, ob das RIM-Objekt die Verbindung zur Datenbank herstellen kann. Führen Sie den Befehl folgendermaßen aus:

  ```
  wrimtest -l RIM-Objektbezeichnung
  ```

  Dabei steht *RIM-Objektbezeichnung* für den vom Inventory Data Handler verwendeten RIM-Objektnamen.

  Anmerkung: Standardmäßig verwendet der Inventory Data Handler lediglich ein RIM-Objekt, und zwar invdh_1. Sie können den Inventory Data Handler jedoch auch mit mehreren RIM-Objekten verbinden. Um die genaue Ursache des Problems festzustellen, müssen Sie diesen Test für jedes vom Inventory Data Handler verwendete RIM-Objekt ausführen.

  Geben Sie die Befehlsoption **x** ein, um das Dienstprogramm **wrimtest** zu verlassen.

  Prüfen Sie ebenfalls, ob die Namen der RIM-Objekte korrekt angegeben sind. Der Name des RIM-Objekts, das während der Installation für den Inventory Data Handler erstellt wird, muss **invdh_1** lauten. Der Name des RIM-Objekts, das für Abfragen verwendet wird, sowie der Name der grafischen Benutzerschnittstelle (GUI) und der Befehle **winvfilter**, **winvrmnode** und **winvsig** muss **inv_query** lauten.

Stellen Sie außerdem sicher, dass alle RIM-Objekte richtig konfiguriert sind. Führen Sie folgenden Befehl für jedes RIM-Objekt aus, um die Konfiguration zu überprüfen:

```
wgetrim RIM-Objektbezeichnung
```

Dabei steht *RIM-Objektbezeichnung* für den Namen des RIM-Objekts. Nähere Informationen dazu, wie Sie die Konfiguration von RIM-Objekten überprüfen, die für die Ausführung mit dem Inventory Data Handler konfiguriert sind, finden Sie unter „RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren” auf Seite 105.

- **SCS wird nicht von dem Profil verwendet.**

  Führen Sie folgende Aktionen aus, um diesen Fehler zu beheben:

  - Prüfen Sie, ob SCS auf allen Gateways und verwalteten Knoten in Ihrer Repeater-Hierarchie installiert ist. Überprüfen Sie zunächst mit dem Befehl **wrpt** die Repeater-Hierarchie. Prüfen Sie anschließend mit dem Befehl **wlsinst**, ob alle Repeater mit SCS installiert worden sind.
  - Prüfen Sie mit dem Befehl **wgetinvdh**, ob der Inventory Data Handler vorhanden ist.
  - Prüfen Sie mit dem Befehl **wgetrim**, ob ein bzw. mehrere RIM-Objekte für die Ausführung mit SCS erstellt worden sind.
  - Prüfen Sie, ob SCS den Plattformtyp des Endpunkts unterstützt.
  - Prüfen Sie, ob SCS den Plattformtyp des Gateways unterstützt. Ist dies nicht der Fall, wird Tivoli Inventory Daten von diesem Gateway nicht mit SCS erfassen.

Anmerkung: Derzeit kann SCS auf Plattformen von verwalteten Knoten installiert werden, die von Solaris, HP-UX, AIX und Windows NT unterstützt werden.

- **Das Depot auf einem Kollektor ist voll.**

  Wenn das Depot auf einem Kollektor voll ist, schreibt der Kollektor für die versuchte Datenerfassung Fehlerinformationen in die CTOC und verschiebt diese in die Fehlerwarteschlange. Von dort wird die CTOC über die Fehlerwarteschlangen der einzelnen Kollektoren in der Hierarchie weitergegeben. Wenn die CTOC am Inventory Data Handler angekommen ist, sendet dieser eine Meldung mit dem Hinweis, dass die Datenerfassung für den betreffenden Endpunkt fehlgeschlagen ist, weil das Depot eines übergeordneten Kollektors voll ist.

  Um diesen Fehler zu beheben, prüfen Sie mit dem Befehl **wcollect** die Größe des Depots. Ist das Depot zu klein, geben Sie die Option **–z** des Befehls **wcollect** ein, um das Depot auf diesem Kollektor, anderen Kollektoren auf der Route und dem Inventory Data Handler zu vergrößern.

- **Ein Kollektor ist fehlgeschlagen.**

  Prüfen Sie, ob die Tivoli Inventory-Hinweisgruppe und die Tivoli Inventory-Statusprotokolldatei Fehlernachrichten enthalten.

  Prüfen Sie, ob eine inaktive Verbindung für den Kollektor gesetzt worden ist. Weitere Informationen zu inaktiven Verbindungen finden Sie unter „Datenerfassung terminieren” auf Seite 109 und „wcollect” auf Seite 247.

  Zeigen Sie mit dem Befehl **wcollect** die Konfiguration des Kollektors an. Prüfen Sie, ob der gesetzte Wert für *Max. Zahl an Wiederholungen* zu niedrig ist. Wenn dieser Wert beispielsweise auf 1 gesetzt ist, wird der Kollektor die Datenerfassung nach dem ersten fehlgeschlagenen Versuch nicht wiederholen.

  Überprüfen Sie die Netzkonnektivität zwischen allen Kollektoren vom Endpunkt zum Inventory Data Handler. Geht die Konnektivität zwischen zwei Kollektoren dieser Route verloren, kann der Inventory Data Handler bis zur Wiederherstellung des Netzwerks keine Fehlernachrichten erhalten.

Um einen fehlgeschlagenen Kollektor wiederherzustellen, müssen Sie den Kollektor mit dem Befehl **wcollect –h** stoppen, sofern er noch aktiv ist. Starten Sie den Kollektor anschließend mit dem Befehl **wcollect –s** erneut.

- **Eine Datenerfassung ist fehlgeschlagen.**

  Führen Sie die gleichen Aktionen aus, die Sie zur Überprüfung eines fehlgeschlagen Kollektors ausführen, prüfen Sie jedoch alle Kollektoren der Route vom Endpunkt zum Inventory Data Handler. Geben Sie zur Bestimmung der Route den Befehl **wrpt** ohne Argumente ein, um die Repeater-Hierarchie festzustellen, und kehren Sie anschließend die Hierarchie um, um die Kollektor-Hierarchie festzustellen.

  Prüfen Sie außerdem die Konnektivität zwischen RIM-Objekten und dem Konfigurations-Repository.

- **Der Statuskollektor gibt fälschlicherweise an, dass ein bereits beendeter Suchlauf noch aktiv ist.**

  Ist während eines Suchlaufs ein nicht wiederherstellbarer Fehler für einen Endpunkt aufgetreten und kann der Statuskollektor hierüber nicht benachrichtigt werden, wird der Statuskollektor davon ausgehen, dass der Suchlauf für diesen Endpunkt noch immer ansteht und nicht beendet ist.

  Überlegen Sie zunächst, ob Sie die Statusinformationen aller Endpunkte aufzeichnen möchten. Je nachdem, welche Optionen Sie gesetzt haben, können Sie möglicherweise keine endgültige Benachrichtigung erhalten. Mit dem folgenden Befehl können Sie die Informationen für den Suchlauf anzeigen:

  ```
  wgetscanstat -a -s -f -p
  ```

  Setzen Sie anschließend den Statuskollektor zurück. Dazu müssen Sie zunächst den Statuskollektorprozess abbrechen. Der Statuskollektorprozess wird auf dem verwalteten Knoten ausgeführt, auf dem sich der Inventory Data Handler befindet, und heißt inv_stat_meths. Mit dem Befehl **wgetinvdh –a** können Sie den Namen für diesen verwalteten Knoten festlegen. Um die ID für den Prozess inv_stat_meths auf UNIX-Systemen festzulegen, führen Sie den Befehl **ps** aus. Verwenden Sie auf Windows

NT-Systemen zur Bestimmung der ID für diesen Prozess den Task-Manager. Sie können ihn durch einen rechten Mausklick auf der Task-Leiste aufrufen.

Verwenden Sie den Befehl **kill**, um den Prozess unter UNIX abzubrechen. Wählen Sie unter NT den gewünschten Prozess auf der Registerkarte 'Prozesse' des Task-Managers aus, und klicken Sie anschließend auf die Schaltfläche **Prozess beenden**.

Löschen Sie schließlich alle gespeicherten Daten für den Suchlauf, der nicht abgeschlossen wurde. Löschen Sie im Verzeichnis $DBDIR/inventory/stat_dir des verwalteten Knotens, auf dem sich der Inventory Data Handler befindet, die Datei, die dem fehlgeschlagenen Suchlauf zugeordnet ist. Der Dateiname lautet inv_stat_*n*.cfg, wobei *n* für die Suchlauf-ID steht.

Legen Sie zur Sicherung der Datenintegrität beim nächsten Suchlauf auf einem Endpunkt, der in der Ausgabe des Befehls **wgetscanstat** als noch anstehend aufgelistet wurde, die Option **Durch aktuelle Ergebnisse ersetzen** für das Profil fest.

- **Die Tivoli Inventory-Hinweisgruppe enthält die Meldung, dass keine Wiederherstellungsoperation ausgeführt werden konnte.**

  Beim Suchlauf speichert der Statuskollektor auf dem Datenträger eine Datei. Wird der Kollektorprozess beispielsweise durch einen Systemabsturz unterbrochen, kann der Status mit Hilfe dieser Datei wiederhergestellt werden. Wenn die gespeicherte Datei beim Neustart des Statuskollektors beschädigt ist, wird in der Tivoli Inventory-Hinweisgruppe ein Fehler aufgezeichnet, der etwa wie folgt lautet:

  ```
  Die Wiederherstellungsoperation konnte für die Datei `/Tivoli/
  deploy/1/users/bob/inv_8333/db/fuji.db/inventory/stat_dir/
  inv_stat_1.cfg' mit der Suchlauf-ID `1' nicht ausgeführt werden.
  Die Datei enthält ungültige Daten.
  ```

Suchen Sie zum Löschen dieses Fehlers im Verzeichnis $DBDIR/inventory/stat_dir des verwalteten Knotens mit dem Inventory Data Handler nach Dateien mit den Namen inv-_stat_*n*.cfg, wobei n eine ganze Zahl ist. Entfernen Sie die Datei, die in der Fehlernachricht angegeben ist. Im obigen Beispiel ist es die Datei inv_stat_1.cfg.

- **Ein anstehender Inventory-Suchlauf bleibt in der Warteschlange des Inventory Data Handler stecken.**

  Zeigen Sie zunächst mit dem folgenden Befehl Informationen über anstehende Suchläufe und Ziele an:

  ```
  wgetscanstat -a -p
  ```

  Prüfen Sie anschließend, ob die Daten in den Warteschlangen des Inventory Data Handler verbleiben, indem Sie die Datendateien der Warteschlange im Verzeichnis $DBDIR/inventory/mc des Inventory Data Handler überprüfen. Weisen die Dateien eine Größe von mehr als 4 Bytes auf, enthalten sie Daten. Weitere Informationen zu diesen Dateien finden Sie unter „Datendateien von Warteschlangen” auf Seite 521.

  Prüfen Sie die Protokolldatei des Inventory Data Handler auf Nachrichten, die darauf hinweisen, dass der Inventory Data Handler zum RIM-Objekt keine Verbindung aufbauen konnte. Möglicherweise werden sinngemäß folgende Nachrichten angezeigt:

  IR: Keine RIM-Objekte gefunden.

  Prüfen Sie, ob die Objekte inv_query und invdh_1 RIM vorhanden sind. Sollte dies nicht der Fall sein, müssen sie erstellt werden. Weitere Informationen finden Sie unter „RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren” auf Seite 105 und in der Beschreibung zum Befehl **wcrtrim** im Handbuch *Tivoli Management Framework Reference Manual*.

IR: FEHLER: Keine gültigen RIM-Objekte für den Inventory Data Handler in zurückgegebener Folge.
Kein RIM-Objekt ist für die Arbeit mit dem Inventory Data Handler konfiguriert. Sie müssen mindestens ein RIM-Objekt zu diesem Zweck konfigurieren. Weitere Informationen finden Sie unter „RIM-Objekte für den Inventory Data Handler erstellen und konfigurieren” auf Seite 105.

IR: FEHLER beim Herstellen einer Verbindung zur Datenbank. Prüfen Sie mit dem Befehl **wrimtest** die Verbindung zu jedem RIM-Objekt, das für die Arbeit mit dem Inventory Data Handler konfiguriert ist.

Halten Sie dann den Inventory Data Handler an, und starten Sie ihn erneut. Verwenden Sie zum Anhalten folgenden Befehl:

```
wcollect -h immediate @InvDataHandler:inv_data_handler
```

Verwenden Sie zum erneuten Starten folgenden Befehl:

```
wcollect -s @InvDataHandler:inv_data_handler
```

- **Ein Suchlauf schlägt mit folgendem Fehler fehl: “Die angegebene Segment-ID (config.dmp) mit Version (*n*) kann nicht entfernt werden, da sie gerade verwendet wird. Versuchen Sie es später noch einmal.”**

  Dieser Fehler tritt gewöhnlich auf, wenn ein verwalteter Knoten von zwei TMRs gemeinsam genutzt wird. Gehen Sie zur Behebung dieses Fehler folgendermaßen vor:

  1. Ändern Sie mit folgendem Befehl das MDist 2-Depotverzeichnis auf dem verwalteten Knoten:

     ```
     wmdist -s managed_node_name rpt_dir=new_directory
     ```

  2. Führen Sie den Objekt-Dispatcher mit folgendem Befehl erneut aus:

     ```
     odadmin reexec
     ```

Abfragen

Folgende Bedingungen können Problem bei Abfragen verursachen:

- Das RIM-Objekt invdh_1 schlägt fehl und verursacht Fehler bei Suchläufen.
- Das RIM-Objekt inv_query schlägt fehl und verursacht Probleme bei Abfragen, bei der grafischen Oberfläche für Inventory-Merkmale oder den Befehlen **winvsig**, **winvfilter** oder **winvrm-node**.

Führen Sie folgenden Befehl aus, um RIM-Objekte auf Fehler bei Suchläufen zu überprüfen.

```
wrimtest -l RIM-Objekt
```

Dabei steht *RIM-Objekt* für den Namen des RIM-Objekts, entweder inv_query oder invdh_1.

Die folgende Nachricht weist auf eine erfolgreiche Verbindung hin. Sie lautet sinngemäß:

```
Reguläre Sitzung wird geöffnet...Sitzung geöffnet
```

Konnte die Verbindung erfolgreich hergestellt werden, geben Sie x ein, um **wrimtest** zu beenden.

Konnte die Verbindung nicht erfolgreich hergestellt werden, wird eine Nachricht ähnlich der folgenden angezeigt:

```
Reguläre Sitzung wird geöffnet...FRWRA0012E Der Aufruf des RDBMS-Servers
ist fehlgeschlagen.
```

Konnte die Verbindung nicht erfolgreich hergestellt werden, prüfen Sie, ob die Kennwörter des RIM-Objekts richtig festgelegt wurden. Überprüfen Sie auch die Ausgabe, die über **wrimtest** generiert wird, auf falsche Einstellungen. Mit dem Befehl **wrimset** können Sie die Einstellungen für jedes RIM-Objekt korrigieren. Sind die Einstellungen korrekt und das Problem bleibt weiterhin bestehen, versuchen Sie mit den Befehlen der RDBMS eine Verbindung zum Konfigurations-Repository herzustellen.

Glossar

A

Abfrage
In einer Tivoli-Umgebung eine Kombination von Anweisungen, mit denen das Konfigurations-Repository nach Systemen durchsucht wird, die bestimmte Kriterien erfüllen.

Abfragebibliothek
In einer Tivoli-Umgebung eine Funktion, mit der Tivoli-Abfragen erstellt und verwaltet werden können.

Abwärts gerichteter Aufruf
In einer Tivoli-Umgebung ein Methodenaufruf, der vom TMR-Server oder Gateway "abwärts gerichtet" an einen Endpunkt erfolgt. Vergleiche hierzu *Aufwärts gerichteter Aufruf*.

Abwärts gerichteter Datenstrom
Gibt die Richtung der Daten an, in die sie in einem Netzwerk fließen.
In einer hierarchischen Netzwerkstruktur eine auf einer untergeordneten Ebene befindliche Netzwerkeinheit. Ein Client beispielsweise befindet sich auf einer tieferen Ebene der Netzwerkstruktur als ein Server.
Vergleiche hierzu *Aufwärts gerichteter Datenstrom*.

admin
Siehe Berechtigungsklasse.

Administrator
Siehe Tivoli-Administrator.

Aufwärts gerichteter Aufruf
In einer Tivoli-Umgebung ein Methodenaufruf, der vom Endpunkt "aufwärts gerichtet" an das Gateway erfolgt. Vergleiche hierzu *Abwärts gerichteter Aufruf*.

Aufwärts gerichteter Datenstrom
Gibt die Richtung an, aus der die Daten in einem Netzwerk kommen.
In einer hierarchischen Netzwerkstruktur eine auf einer übergeordneten Ebene befindliche Netzwerkeinheit. Ein Server beispielsweise befindet sich auf einer höheren Ebene der Netzwerkstruktur als ein Client.
Vergleiche hierzu *Abwärts gerichteter Datenstrom*.

Ausfüllen
In der Tivoli-Umgebung das Ausfüllen eines Profils mit den Informationen, die an die zugeordneten verwalteten Ressourcen verteilt werden.

B

Berechtigungsklasse
Siehe Berechtigungsklasse.

Berechtigungsklasse
Tivoli-Administratoren verfügen über eine bestimmte Berechtigungsklasse, die die Ausführung zugewiesener Systemmanagement-Tasks ermöglicht. Die Berechtigungsklasse kann für die gesamte Tivoli Management Region (TMR) oder für bestimmte Ressourcen, wie z. B. die, die in einem Richtlinienbereich enthalten sind, erteilt werden. Beispiele für Berechtigungsklassen sind: super, senior, admin und user.

bestehen bleiben
Sitzungsübergreifend speichern, in der Regel in einem permanenten Speicher wie etwa einem Datenbanksystem oder Verzeichnis.

E

Endpunkt
(1) In einer Tivoli-Umgebung ein Tivoli-Client, der letztendlich der Empfänger von Tivoli-Operationen ist. (2) In einer Tivoli-Umgebung ein Tivoli-Dienst, der auf mehreren Betriebssystemen aktiv ist und auf diesen Systemen Tivoli-Operationen ausführt, die es Tivoli Management Framework ermöglichen, diese Systeme als Tivoli-Clients zu verwalten. (3) In Tivoli IT Director ein verwaltetes System oder Gerät.

Ereigniskonsole
Eine grafische Benutzerschnittstelle von Tivoli Enterprise Console, mit der Systemadministratoren vom Ereignis-Server versendete Ereignisse anzeigen und auf diese antworten können. Tivoli Event Integration Facility wirkt sich weder direkt auf die Ereigniskonsole aus, noch verwendet es diese direkt.

Ereignisserver
Im Produkt Tivoli Enterprise Console ein zentraler Server, der Ereignisse verarbeitet. Der Ereignisserver erstellt einen Eintrag für jedes ankommende Ereignis und wertet das Ereignis anhand einer Regelbasis aus, um zu ermitteln, ob er automatisch auf dieses Ereignis reagieren oder es automatisch ändern kann. Der Ereignisserver aktualisiert außerdem die Ereigniskonsolen mit den aktuellen Ereignisinformationen. Ist der primäre Ereignisserver nicht verfügbar, können Ereignisse an einen sekundären Ereignisserver gesendet werden.

G

Gateway
In einer Tivoli-Umgebung die Software, die auf einem verwalteten Knoten ausgeführt wird und den gesamten Kommunikations-Dienst zwischen einer Gruppe von Endpunkten und der sonstigen Tivoli-Umgebung verwaltet. Das Gateway verfügt

auch über eine integrierte Multiplexverteilungsfunktion (MDist 2) und kann so als Ausgabefächerungspunkt Verteilungen an eine Vielzahl von Endpunkten vornehmen.

H

Hinweis
In der Tivoli-Umgebung eine von einer Systemmanagementoperation generierte Nachricht, die Informationen zu einem Ereignis oder dem Status einer Anwendung enthält. Die Hinweise werden in Hinweisgruppen gespeichert.

Hinweisgruppe
In der Tivoli-Umgebung ein anwendungs- bzw. operationsspezifischer Container, in dem Hinweise zu bestimmten Tivoli-Funktionen gespeichert und angezeigt werden. Das Schwarze Brett von Tivoli besteht aus Hinweisgruppen. Einem Tivoli-Administrator können eine oder mehrere Hinweisgruppe(n) zugeordnet werden; das Schwarze Brett des Administrators enthält nur die Hinweise aus einer Hinweisgruppe, die dem Administrator zugeordnet wurde.

Host
In einer Tivoli-Umgebung ein Computer, der bei der Profilverteilung als verwalteter Knoten dient.

I

IDL
Siehe Interface Definition Language.

Interface Definition Language (IDL)
(Schnittstellendefinitionssprache) In CORBA eine deklarative Programmiersprache, die zur Beschreibung von Objektschnittstellen ohne Berücksichtigung der Objektimplementierung verwendet wird.

Inventory Data Handler
In einer Scalable Collection Service-Topologie das Tivoli Inventory-Objekt, das Daten aus einem Inventory-Suchlauf empfängt und mindestens eine Verbindung zum Senden dieser Daten an das Konfigurations-Repository verwendet.

J

Job
Eine Ressource, die aus einer Task und deren vorkonfigurierten Ausführungsparametern besteht. Die Ausführungsparameter legen unter anderem die Gruppe der Hosts fest, auf denen der Job ausgeführt werden soll.

K

Klonen
In der Tivoli-Umgebung eine Operation, bei der ein Tivoli-Administrator Profile replizieren kann. Diese Funktion vereinfacht die Erstellung mehrerer Profile mit ähnlichen Merkmalen. *Siehe auch* Profilprototyp (prototype file).

Kollektor
In einer Tivoli-Umgebung entweder (a) eine Repeater-Station, auf der der SCS-Dienst (Scalable Collection Service) installiert ist, oder b) ein SCS-Dämon auf einem verwalteten Knoten oder Gateway, auf dem Daten gespeichert sind und dann an andere Kollektoren oder den Inventory Data Handler weitergeleitet werden.

Konfigurations-Repository
In einer Tivoli-Umgebung ein RIM-Repository, das Informationen enthält, die von Tivoli Inventory und Tivoli Software Distribution erfasst bzw. generiert und gespeichert werden. Tivoli Inventory speichert beispielsweise Informationen zu Hardware, Software, Systemkonfiguration und physischem Inventar, während Tivoli Software Distribution Informationen zu Softwarepaketen und Dateipaketoperationen speichert.

L

Lokale Verteilung
In einer Tivoli-Umgebung eine Verteilung an Zielmaschinen, die sich in derselben TMR wie die Quellenmaschine befinden.

M

Management Information Format (MIF)
Die DMI-Spezifikation (Desktop Management Interface), die die Syntax zur Beschreibung von Verwaltungsinformationen über Hardware- und Softwarekomponenten, die auf einem Computer installiert sein können, definiert.

MDist 2
Ein Multiplexverteilungsdienst von Tivoli Management Framework, der eine effiziente Übertragung von Daten an viele Ziele ermöglicht. Administratoren können damit den gesamten Verteilungszyklus überwachen und steuern. Der Multiplexverteilungsdienst MDist hingegen weist diese Verwaltungsfunktionen nicht auf.

Methode
In der objektorientierten Entwicklung bzw. Programmierung die Software, die die Ausführung eines angegebenen Vorgangs übernimmt.

MIF
Siehe Management Information Format.

N

Namensregistrierdatenbank
Ein Namensdienst, der aus einer zweidimensionalen Tabelle besteht, mit deren Hilfe in einer Tivoli Management Region (TMR) Ressourcennamen zu Ressourcen-IDs und zugehörigen Informationen zugeordnet werden.

Network Computing
Der Einsatz einer Infrastruktur zur skalierbaren verteilten Datenverarbeitung, die sich die Schlüsselelemente der modernen Netztechnologie wie beispielsweise System- und Netzverwaltung, Internet, Intranets, Clients und Server, Anwendungsprogramme, Datenbanken, Transaktionsverarbeitung sowie die Verwendung verschiedener Betriebssysteme und Kommunikationsprotokolle zunutze macht.

O

Object Request Broker (ORB)
In der objektorientierten Programmierung die Software, die als Schnittstelle dient, indem sie auf transparente Weise den Austausch von Anforderungen und Antworten zwischen Objekten ermöglicht.

Objektgruppe
In der Tivoli-Umgebung ein Container, der Objekte auf einer Tivoli-Arbeitsoberfläche zu Gruppen zusammenfasst und so dem Tivoli-Administrator eine einzige Ansicht zusammengehöriger Ressourcen bietet. Objektgruppen können entweder von Tivoli Management Framework oder einem Tivoli-Administrator erstellt werden. Die Bestandteile einer Objektgruppe werden als Mitglieder bezeichnet. Beispiele für Objektgruppen sind die Administratorobjektgruppe, die generische Objektgruppe und die Monitorobjektgruppe; die Administratorobjektgruppe ist ein Beispiel für eine von Tivoli Management Framework erstellte Objektgruppe.

Objektpfad
Ein absoluter oder relativer Pfad eines Tivoli-Objekts, ähnlich dem Pfad in einem Dateisystem.

Objektverweis
Eine Objekt-ID, die einem Objekt während seiner Erstellung zugewiesen wird.

OID
Objekt-ID

ORB
Siehe Object Request Broker.

oserv

Der Name des Object Request Broker (ORB), der in der Tivoli-Umgebung verwendet wird. Oserv wird auf dem Tivoli Management Region-Server und allen Tivoli Management Region-Clients ausgeführt.

P

permanent

Daten, die sitzungsübergreifend gespeichert werden, in der Regel in einem permanenten Speicher wie etwa einem Datenbanksystem oder Verzeichnis.

Profil

In einer Tivoli-Umgebung ein Container für anwendungsspezifische Informationen zu einer bestimmten Ressourcenart. Eine Tivoli-Anwendung gibt die Schablone für ihre Profile an, die Informationen zu den Ressourcen enthält, die von der Tivoli-Anwendung verwaltet werden können.
Ein Profil wird im Zusammenhang mit einem Profilmanager erstellt, der das Profil mit einer Tivoli-Ressource, z. B. einem verwalteten Knoten, verbindet, die die in dem Profil enthaltenen Informationen verwendet. Ein Profil verfügt über keine direkten Subskribenten.

Profilmanager

Ein Behälter für Profile, der die Profile mit Ressourcen verbindet, die als “Subskribenten” bezeichnet werden. Tivoli-Administratoren verwenden Profilmanager zum Ordnen und Verteilen von Profilen. Ein Profilmanager wird in einem Richtlinienbereich erstellt und stellt darin eine verwaltete Ressource dar. Ein Profilmanager kann in folgenden Modi arbeiten: Datenloser Modus: Die Verteilung durch den Profilmanager kann an beliebige Tivoli-Clients (einschließlich verwaltete Knoten, Endpunkte und NetWare-verwaltete Stationen), jedoch nicht an andere Profilmanager erfolgen. Datenbankmodus: Die Verteilung durch den Profilmanager kann an beliebige Profilmanager (im datenlosen Modus oder Datenbankmodus), alle verwalteten Knoten und NetWare-verwaltete Stationen, jedoch nicht an Endpunkte erfolgen.

Profilprototyp

Ein Profilmodell, das von Tivoli-Administratoren als Vorlage zur Erstellung anderer Profile verwendet werden kann. Oft geschieht dies durch Klonen des Profilprototyps.

R

RDBMS Interface Module (RIM)

In Tivoli Management Framework das Modul in der verteilten Objektdatenbank, die Informationen über die Installation des Verwaltungssystems für relationale Datenbanken enthält.

Registrierter Name
Der Name, unter dem eine bestimmte Ressource bei ihrer Erstellung in der Namensregistrierdatenbank registriert wird.

Repeater-Station
In einer TMR ein verwalteter Knoten, der über die Funktion MDist oder MDist 2 verfügt. Eine Repeater-Station erhält eine einzige Kopie von Daten und verteilt sie an die nächste Ebene von Clients.

Ressource
Siehe Verwaltete Ressource.

Ressourcenart
Eines der Merkmale einer verwalteten Ressource. Ressourcenarten werden in der Standardwertegruppe eines Richtlinienbereichs definiert.

Richtlinie
Auf verwaltete Ressourcen angewendete Regeln. Eine spezielle Regel wird als "Richtlinienmethode" bezeichnet.

Richtlinienbereich
In einer Tivoli-Umgebung eine Gruppe verwalteter Ressourcen, denen eine oder mehrere Richtlinien gemeinsam sind. Tivoli-Administratoren verwenden Richtlinienbereiche dazu, die Management- und Organisationsstruktur einer Network Computing-Umgebung zu gestalten. Administratoren können ähnliche Ressourcen zu Gruppen zusammenfassen, den Zugriff auf Ressourcen definieren, Ressourcen steuern sowie Regeln zur Verwaltung der Ressourcen zuordnen. Ein Richtlinienbereich umfasst Ressourcenarten und eine Liste der zu verwaltenden Ressourcen. Richtlinienbereiche werden auf der Tivoli-Arbeitsoberfläche durch ein Symbol gekennzeichnet, das ein typisches amerikanisches Stadt- oder Staatsverwaltungsgebäude (Kuppelbau mit Säulenfront) zeigt. Bei der Erstellung einer Tivoli Management Region (TMR) wird auch ein Richtlinienbereich mit demselben Namen erstellt. In diesem Fall verfügt die TMR nur über einen Richtlinienbereich. Normalerweise erstellen Tivoli-Administratoren jedoch weitere Richtlinienbereiche und -unterbereiche, mit denen die Organisation der TMR dargestellt wird. Eine TMR bezieht sich auf die tatsächliche physische Konnektivität, ein Richtlinienbereich dagegen auf die logische Anordnung der Ressourcen.

Richtlinien für Gültigkeitsprüfung
In einer Tivoli-Umgebung eine Richtlinie, die sicherstellt, dass alle Ressourcen in einem Richtlinienbereich die festgelegten Richtlinien des Bereichs erfüllen. Durch die Richtlinie für Gültigkeitsprüfung wird verhindert, dass Tivoli-Administratoren Ressourcen erstellen und bearbeiten, die nicht den Richtlinien des Bereichs entsprechen, in dem die Ressourcen erstellt wurden.

Richtlinienunterbereich
In einer Tivoli-Umgebung ein Richtlinienbereich, der innerhalb eines übergeordneten Richtlinienbereichs erstellt oder gespeichert wird. Wenn ein Richtlinienunterbereich erstellt wird, benutzt er zuerst die Ressource und die Richtlinienmerkmale des übergeordneten Richtlinienbereichs. Tivoli-Administratoren können zu einem späteren Zeitpunkt diese Merkmale an die besonderen Bedürfnisse und Unterschiede des Unterbereichs anpassen.

RIM
Siehe RDBMS Interface Module.

RIM-Host
Der verwaltete Knoten, auf dem mindestens ein RIM-Objekt installiert ist. *Siehe auch* RIM-Objekt.

RIM-Objekt
Das Objekt, von dem aus die Datenbank-Client-Software durch den Tivoli-Administrator ausgeführt und eine Verbindung zum Server des Verwaltungssystems für relationale Datenbanken (RDBMS) hergestellt wird. *Siehe auch* RIM-Host.

S

Scalable Collection Service (SCS)
Ein Service, der eine effektive, asynchrone Erfassung großer Datenmengen in mehreren komplexen Netzwerken ermöglicht.

Schema
Eine Gruppe von Anweisungen in einer Datendefinitionssprache, die eine vollständige Beschreibung der Datenbankstruktur liefern.

SCS
Siehe Scalable Collection Service (SCS).

senior
Siehe Berechtigungsklasse.

Signatur
In der Computersoftware der Name einer Operation und deren Parameter.

Skalierbar
Bezieht sich auf die Fähigkeit eines Systems, sich schnell an eine Steigerung oder Verringerung der Auslastung, des Datenbestands oder der Anforderungen anzupassen. Ein skalierbares System kann sich beispielsweise effizient an die Arbeit mit größeren oder kleineren Netzen zur Ausführung von Tasks unterschiedlicher Komplexität anpassen.

Standardwertegruppe
Eine Gruppe von Merkmalwerten, die einer Ressource bei ihrer Erstellung zugeordnet werden.

Subskribent
In einer Tivoli-Umgebung ein verwalteter Knoten, ein Profilmanager, ein Endpunkt oder ein anderer Tivoli-Client, der Subskribent eines Profilmanagers ist. Obwohl Profile an Subskribenten verteilt werden, sind die Subskribenten nicht immer der endgültige Zielort der verteilten Profile.

Subskription
In einer Tivoli-Umgebung der Prozess, mit dem die Subskribenten für die Profilverteilung festgelegt werden.

Subskriptionsliste
In einer Tivoli-Umgebung eine Liste, mit der die Subskribenten einem Profilmanager zugeordnet werden. Wenn einer Subskriptionsliste (eigentlich eine Liste innerhalb einer anderen Liste) ein Profilmanager hinzugefügt wird, werden mehrere Ressourcen gleichzeitig der Subskriptionsliste hinzugefügt und müssen nicht einzeln hinzugefügt werden. In Tivoli Plus-Modulen dient ein Profilmanager als Subskriptionsliste.

Suchlauf
Ein Vorgang, bei dem Inventarinformationen erfasst werden.

super
Siehe Berechtigungsklasse.

T

Task
Die Definition einer Aktion, die routinemäßig auf verschiedenen verwalteten Knoten im gesamten Netzwerk ausgeführt werden muss. Eine Task definiert die für die Task erforderlichen ausführbaren Dateien, die für die Ausführung der Task erforderliche Berechtigungsklasse sowie den Namen des Benutzers bzw. der Gruppe, unter dem bzw. der die Task ausgeführt wird.

Tivoli-Administrator
In einer Tivoli-Umgebung ein Systemadministrator, der berechtigt ist, Systemmanagementaufgaben auszuführen und Richtlinienbereiche in einem oder mehreren Netzwerken zu verwalten. Für jeden Tivoli-Administrator wird auf der Arbeitsoberfläche ein Symbol angezeigt.

Tivoli-Arbeitsoberfläche
Die Arbeitsoberfläche, die Systemadministratoren in der Tivoli-Umgebung zur Verwaltung ihrer Network Computing-Umgebung verwenden.

Tivoli-Client

In einer Tivoli-Umgebung ein beliebiger Computer, mit Ausnahme des TMR-Servers, auf dem Tivoli Management Framework installiert ist. TMR-Clients verfügen über eine lokale Objektdatenbank, und der Dämon 'oserv' kommt auf ihnen zur Ausführung.

Tivoli Enterprise Console

Ein Tivoli-Produkt, das Fehlerberichtigungen für System-, Anwendungs-, Netzwerk- und Datenbankereignisse ermittelt, verarbeitet und automatisch auslöst. Es ist der zentrale Steuerpunkt für Ereignisse von allen Quellen. Tivoli Enterprise Console bietet von zentraler Stelle aus einen umfassenden Überblick über die Network Computing-Umgebung; TEC verwendet verteilte Ereignismonitore zum Erfassen von Informationen, einen zentralen Ereignisserver zum Verarbeiten von Informationen und verteilte Ereigniskonsolen zum Darstellen von Informationen für Systemadministratoren.

Tivoli Inventory

Ein Tivoli-Produkt, mit dem Systemadministratoren Hardware- und Softwareinformationen zu einer Network Computing-Umgebung erfassen können. Es durchsucht die verwalteten Ressourcen und speichert Inventarinformationen im Konfigurations-Repository.

Tivoli Management Region (TMR)

Ein TMR-Server und die ihm zugeordneten Clients. Eine Organisation kann über mehrere TMRs gleichzeitig verfügen. Werden verbundene TMRs verwendet, ist dies für den Benutzer erkennbar.

Anmerkung: Eine TMR bezieht sich auf die tatsächliche physische Anordnung, ein Richtlinienbereich dagegen auf die logische Struktur der Ressourcen.

Tivoli Software Distribution

Ein Tivoli-Produkt, das die Softwareverteilung an Clients und Server in einer Network Computing-Umgebung automatisiert. Eine Organisation kann dieses Produkt zum koordinierten und konsistenten Installieren und Aktualisieren von Software in einem Netz verwenden. Tivoli Software Distribution erstellt Softwarepakete (in Version 4) und Dateipakete (in Version 3) und verteilt sie an vordefinierte Subskribenten.

Tivoli-Umgebung

Die auf dem Tivoli Framework basierenden Tivoli-Anwendungen, die bei einem bestimmten Kunden installiert sind und dem plattformübergreifenden Netzwerkmanagement dienen. Innerhalb einer Tivoli-Umgebung kann der Systemadministrator Softwarepakete verteilen, Benutzerkonfigurationen verwalten, Zugriffsberechtigungen ändern, Arbeitsgänge automatisieren, Ressourcen überwachen sowie Zeitpläne für bestimmte Aufgaben erstellen.

TMR

Siehe Tivoli Management Region.

TMR-Server

Der Server, auf dem das gesamte Tivoli-Softwarepaket gespeichert ist bzw. der auf dieses verweist, einschließlich der kompletten Objektdatenbank.

U

Umfassende Verwaltungsumgebung

Der Tivoli-Ansatz zur Erstellung verwaltbarer Anwendungen und zum Einsatz bzw. zur Synchronisation von Anwendungen und Systemkonfigurationen auf Unternehmensebene.

user

Siehe Berechtigungsklasse.

V

Verwaltete Ressource

In der Tivoli-Umgebung jede Hardware- oder Softwareentität (Maschine, Dienst, System oder Einrichtung), die durch ein Datenbankobjekt und ein Symbol auf der Tivoli-Arbeitsoberfläche dargestellt wird. Verwaltete Ressourcen müssen eine in einem Richtlinienbereich unterstützte Ressourcenart haben und unterliegen einem Regelsatz. Verwaltete Ressourcen umfassen verwaltete Knoten, Task-Bibliotheken, Monitore, Profile, Schwarze Bretter u.a.

Verwalteter Knoten

In einer Tivoli-Umgebung eine verwaltete Ressource, auf der Tivoli Management Framework installiert ist.

Verwaltung durch Subskription

In der Tivoli-Umgebung das Konzept zur Verwaltung von Netzwerkressourcen durch die Erstellung von Profilen, die über Profilmanager an physische Entitäten (Tivoli-Ressourcen), die so genannten Subskribenten, verteilt werden.

Z

Ziel

In der Tivoli-Umgebung ein Tivoli-Client, auf dem ein Job oder andere Vorgänge ausgeführt werden.

Index

Sonderzeichen

$BINDIR/TME/INVENTORY, Verzeichnis 187
$DBDIR/inventory/stat_dir, Verzeichnis 259
$DBDIR/mcollect, Verzeichnis 249
$DBDIR/mcollect/depot directory 527

Numerische Stichwörter

128MB_SUBSCRIPTIONS 437
256MB_SUBSCRIPTIONS 437
512MB_SUBSCRIPTIONS 437

A

Abfrage ausführen, Dialog 222
Abfrage erstellen, Dialog 211, 213
Abfragebibliotheken, erstellen 213
Abfrageergebnisse exportieren, Dialog 222
Abfragen
  anpassen 215
  ausführen 209, 221, 224
  Beispiel 212, 220
  Berechtigungsklassen 237
  erstellen 211, 220
  Erstellen von Abfragen 213
  Fehlerbehebung 535
  im TMR-Verbund 233
  standardmäßig in Tivoli Inventory
    erstellen 87
    Liste 415, 439
  Subskribenten ermitteln 226
  über das Symbolmenü ausführen 231
  über die Befehlszeilenschnittstelle (CLI) ausführen 318
  Übersicht 209
Abfragen *(Forts.)*
  Ziele ermitteln 225
Abfragen, Inventory
  CDROM_QUERY 415
  COMPUTER_MEM_QUERY 416
  FLPY_DRV_QUERY 416
  H_CDROM_QUERY 435
  H_COMPUTER_MEM_QUERY 435
  H_COMPUTER_QUERY 435
  H_FLPY_DRV_QUERY 435
  H_HDISK_QUERY 435
  H_HEADER_INFO_QUERY 435
  H_INST_FILE_QUERY 435
  H_INVENTORY_SWARE 435
  H_IP_ADDR_QUERY 435
  H_IPX_ADDR_QUERY 435
  H_MOUSE_QUERY 435
  H_NATIV_SWARE_QUERY 435
  H_NET_CARD_QUERY 435
  H_NOSIG_FILES_QUERY 436
  H_NW_SERVER_QUERY 436
  H_NW_VOLS_QUERY 436
  H_OS_QUERY 436
  H_PARTITION_QUERY 436
  H_PC_BIOS_QUERY 436
  H_PCI_DEV_QUERY 436
  H_PCPROCESSOR_QUERY 436
  H_PRINTER_QUERY 436
  H_PROCESSOR_QUERY 436
  H_STORAGE_DEV_QUERY 436
  H_TAPEDRV_QUERY 436
  H_UNIX_SYS_QUERY 436
  H_USB_DEV_QUERY 436
  H_VID_CARD_QUERY 436
  HDISK_QUERY 417
  HEADER_INFO_QUERY 418
  INST_FILE_QUERY 418
  INVENTORY_HWARE 419
  INVENTORY_SWARE 420

Abfragen, Inventory *(Forts.)*
IP_ADDR_QUERY 420
IPX_ADDR_QUERY 421
KEYBOARD_QUERY 421
MATCH_CRC32_QUERY 422
MATCH_MD5_QUERY 422
MATCH_QUICK_QUERY 423
MOUSE_QUERY 424
NATIV_SWARE_QUERY 424
NET_CARD_QUERY 425
NOSIG_FILES_QUERY 425
NW_SERVER_QUERY 426
NW_VOLS_QUERY 427
OS_QUERY 427
PARTITION_QUERY 428
PC_BIOS_QUERY 428
PC_PROCESSOR_QUERY 429
PCI_DEV_QUERY 430
PRINTER_QUERY 431
PROCESSOR_QUERY 431
STORAGE_DEV_QUERY 432
TAPEDRV_QUERY 432
UNIX_SYS_QUERY 433
USB_DEV_QUERY 433
VID_CARD_QUERY 434
Abfragen, Subskription
128MB_SUBSCRIPTIONS 437
256MB_SUBSCRIPTIONS 437
512MB_SUBSCRIPTIONS 437
AIX_SUBSCRIPTIONS 437
H_AIX_SUBSCRIPTIONS 438
H_HPUX_SUBSCRIPTIONS 438
H_NETWARE_SUBSCRIPTIONS 438
H_OS2_SUBSCRIPTIONS 438
H_Solaris_SUBSCRIPTIONS 438
H_WIN_2000_SUBSCRIPTIONS 438
H_WIN_98_SUBSCRIPTIONS 439
H_WIN_ALL_SUBSCRIPTIONS 439
H_WIN_ME_SUBSCRIPTIONS 439
H_WIN_NT_SUBSCRIPTIONS 439
HPUX_SUBSCRIPTIONS 437
NETWARE_SUBSCRIPTIONS 437
OS2_SUBSCRIPTIONS 437
Solaris_SUBSCRIPTIONS 437
SUB_128MB_SUBSCRIPTIONS 437
WIN_2000_SUBSCRIPTIONS 437
Abfragen, Subskription *(Forts.)*
WIN_98_SUBSCRIPTIONS 437
WIN_ALL_SUBSCRIPTIONS 437
WIN_ME_SUBSCRIPTIONS 437
WIN_NT_SUBSCRIPTIONS 438
Absoluter Pfad 242
ACCESSED_TIME, Beschreibung 377
ADAPTER_MODEL, Beschreibung 388
ADAPTER_TYPE, Beschreibung 388
admin, Berechtigungsklasse 236
Administrator erstellen, Dialog 131
Administrator-Ressourcenarten 169
AIX_SUBSCRIPTIONS 437
Anmeldenamen festlegen, Dialog 132
anpassen
Konfigurations-Repository 135, 158
Softwaresuchläufe 174
Anpassen
Abfragen 211, 220
Hardwaresuchläufe 183, 189, 337, 354
Profile 171, 191
Softwaresuchläufe 183, 342, 358
Ansichten
CDROM_VIEW 370
COMPUTER_MEM_VIEW 372
COMPUTER_VIEW 373
FLPY_DRV_VIEW 374
H_CDROM_VIEW 414
H_COMPUTER_VIEW 414
H_FLPY_DRV_VIEW 414
H_HDISK_VIEW 414
H_HEADER_VIEW 414
H_INST_FILE_VIEW 414
H_INST_SWARE_VIEW 414
H_IP_ADDR_VIEW 414
H_IPX_ADDR_VIEW 414
H_MEM_VIEW 414
H_MOUSE_VIEW 414
H_NATIV_VIEW 414
H_NET_CARD_VIEW 414
H_NOSIG_FILES_VIEW 414
H_NW_SERVER_VIEW 414
H_NW_VOLS_VIEW 414
H_OS_VIEW 414
H_PARTITION_VIEW 414
H_PC_BIOS_VIEW 414

Ansichten *(Forts.)*
H_PCI_DEV_VIEW 415
H_PCPROCESSOR_VIEW 415
H_PRINTER_VIEW 415
H_PROCESSOR_VIEW 415
H_STORAGE_DEV_VIEW 415
H_TAPEDRV_VIEW 415
H_UNIX_SYS_VIEW 415
H_USB_DEV_VIEW 415
H_VID_CARD_VIEW 415
HDISK_VIEW 375
HEADER_INFO_VIEW 376
INST_FILE_VIEW 377
INST_SWARE_VIEW 379
INVENTORYDATA 380
IP_ADDR_VIEW 381
IPX_ADDR_VIEW 383
KEYBOARD_VIEW 384
MATCH_SWARE_VIEW 385
MOUSE_VIEW 386
NATIV_SWARE_VIEW 387
NET_CARD_VIEW 388
NOSIG_FILES_VIEW 389
NW_SERVER_VIEW 390
NW_VOLS_VIEW 393
OS_VIEW 394
PARTITION_VIEW 395
PC_BIOS_VIEW 397
PC_PROCESSOR_VIEW 398
PCI_DEV_VIEW 402
PRINTER_VIEW 403
PROCESSOR_VIEW 404
Protokoll 413, 415
Standard, Liste 370, 415
STORAGE_DEV_VIEW 405
SWARE_MATCH_CRC32 406
SWARE_MATCH_MD5 407
SWARE_MATCH_QUICK 409
USB_DEV_VIEW 410
VID_CARD_VIEW 411
Anwendungen, Zusammenspiel 16
Anwendungskennung 105
Anzeigen
benutzerspezifische 156
hinzufügen 156
Auswählen
RDBMS-Server 31
RIM-Host 30

B

backup, Berechtigungsklasse 236
BasicInventoryConfig-Richtlinienobjekt 465
Befehle
dmiscan 514
für SCS 245
für Tivoli Inventory 243
idlcall 108
odadmin 23, 111, 520, 534
Syntax 240
tail 522, 523, 524, 525
Übersicht 239
wadminep 512, 513, 528
wbkupdb 38
wcollect 23, 95, 101, 111, 115, 247, 522, 534
wcomprules 61
wcprb 61
wcrtadmin 134
wcrtinvcb 104, 256
wcrtinvdh 100, 103, 118, 258, 274, 276, 321
wcrtpol 466
wcrtprf 193
wcrtquery 143
wcrtrb 61
wcrtrim 51, 105
wcstat 115, 253, 262, 522
wdel 103, 104, 195
wdistinv 204, 265
wepscan 204, 271, 516
wgetinvdh 116, 274, 529
wgetinvglobal 117, 119, 277
wgetinvpcfiles 282
wgetinvpchw 285
wgetinvpcsw 288
wgetinvunixfiles 294
wgetinvunixhw 297
wgetinvunixsw 300
wgetpolm 467

Befehle *(Forts.)*
wgetrim 32, 529
wgetscanstat 117, 119, 306, 527, 528
wimprbclass 61
winstall 52
winvfilter 308
winvrmnode 142, 311
winvsig 124, 125, 313
wloadrb 61
wlookup 107
wlsinst 529
wlspolm 467
wmdist 202
wpatch 29
wputpolm 468
wqueryinv 318
wrimset 535
wrimtest 528, 535
wrimtrace 526
wrpt 95, 96, 252, 529, 531
wruninvquery 228
wrunquery 221, 224, 228
wsetadmin 134
wsetinvdh 100, 118, 259, 274, 276, 321
wsetinvglobal 118, 259, 261, 275, 278, 322, 325, 524
wsetinvpcfiles 331
wsetinvpchw 337
wsetinvpcsw 342
wsetinvunixfiles 348
wsetinvunixhw 354
wsetinvunixsw 358
wsetpr 469
wsetrim 32, 108
wsetrimpw 32, 52, 66, 70, 83
wstartesvr 61
wstopesvr 61
wuninst 57
Benachrichtigung
siehe Statusinformationen 100
Benachrichtigung über Suchläufe
siehe Statusinformationen 100
benennen
MIF-Benutzerdateien 159
Tabellen und Spalten für DB2- und Informix- 138
Benennen
Abfragebibliotheken 88
Abfragen 88
MIF-Benutzerdateien 137
Profile 166
Tabellen und Spalten 136
Tabellen und Spalten für DB2- und Informix- 136
Benutzerdatenformular 152, 154
Benutzerdatenschablone
MIF-Attribute hinzufügen 146, 152
MIF-Gruppen hinzufügen 146, 152
Seite 146
Benutzerinformationen
abrufen und sichern 156
anzeigen 156
erfassen 127, 155
Berechtigungsklassen 17
Abfragen
eigene erstellen 212
Liste mit Berechtigungsklassen 237
Standardabfragen erstellen 88
Subskribenten ermitteln 225
Abfragen ausführen 221
admin, Berechtigungsklasse 236
backup, Berechtigungsklasse 236
für Tivoli Inventory erforderlich 236
Install_client, Berechtigungsklasse 236
Install_product, Berechtigungsklasse 236
Installation
Anwendung 39
Softwaresignaturen 86
Tivoli Inventory-Abfragen 88
Inventory-Daten anzeigen 229
Inventory_end_user, Berechtigungsklasse 235
Inventory_query, Berechtigungsklasse 235
Inventory_scan, Berechtigungsklasse 235
Inventory_view, Berechtigungsklasse 235
Profile
anpassen 191
erstellen 166
klonen 192
löschen 194
verteilen 199
Query_edit, Berechtigungsklasse 235

Berechtigungsklassen 17 *(Forts.)*
Query_execute, Berechtigungsklasse 235
Query_view, Berechtigungsklasse 235
restore, Berechtigungsklasse 236
Richtlinie
ersetzen 467
erstellen 466
zuweisen 469
RIM_update, Berechtigungsklasse 145, 235
RIM_view, Berechtigungsklasse 145, 235
senior, Berechtigungsklasse 236
Softwaresignaturen 124
super, Berechtigungsklasse 236
user, Berechtigungsklasse 236
Betriebsdatentabellen 366
BIOS_DATE, Beschreibung 397
BIOS_ID, Beschreibung 397
BIOS_ID_BYTES, Beschreibung 397
bios.ini 162
BIOS_MANUFACTURER, Beschreibung 397
BIOS_MODEL, Beschreibung 397
BIOS_SER_NUM, Beschreibung 397
BIOS_STRING, Beschreibung 397
Bündel
Bündelung aktivieren 101, 275, 322
das Sendeintervall konfigurieren 100
Informationen anzeigen 275
maximale Anzahl von Zielen konfigurieren 101, 260, 322
Sendeintervall konfigurieren 259, 322
Sendeziel konfigurieren 327
Zieladresse konfigurieren 101
BUTTONS, Beschreibung 386

C

Callback-Objekt von Inventory
Definition 6
erstellen 256
Host auswählen 45
verschieben 103
CDROM_QUERY 415
CDROM_VIEW 370
checkpointGL_dqfile.dat 521
checkpointGL_eqfile.dat 521
checkpointGL_iqfile.dat 521
checkpointGL_oqfile.dat 521
CHECKSUM_CRC32, Beschreibung 378
CHECKSUM_MD5, Beschreibung 378
CHECKSUM_QUICK, Beschreibung 378
CHIP_FAMILY, Beschreibung 398
CHIP_MODEL, Beschreibung 398
CHIP_STEPPING, Beschreibung 398
CMPXCHG8B_SUPP, Beschreibung 399
Collection Table of Contents 4
COMPUTER, Tabelle 440
COMPUTER_MEM_QUERY 416
COMPUTER_MEM_VIEW 372
COMPUTER_MODEL, Beschreibung 373
COMPUTER_SCANTIME, Beschreibung 373
COMPUTER_SYS_ID, Beschreibung 371, 372
COMPUTER_SYS_MEM, Tabelle 441
COMPUTER_VIEW 373
COND_MOVE_SUPP, Beschreibung 399
config.dmp 272
CRC (Cyclical Redundancy Check) 344
CREATED_TIME, Beschreibung 377
CTOC
Aufgabenbereich bei der Datenerfassung 93, 94
Beschreibung 4
Funktion bei der Datenerfassung 10, 11
Informationen anzeigen 263
Informationen anzeigen zu 115
CTOC_log.dat 525
CURRENT_ADDR, Beschreibung 388

D

Das Fenster ’Signaturen’ 124
Datei-Browser, Dialog 26
Dateierweiterungen, Richtlinie 487, 501
Dateinamen, Regeln für deren Angabe 333, 350
Dateneinheiten 98, 113, 250
Datenerfassung
Fehlerbedingungen 522
fehlgeschlagen 531
mit SCS 9

Datenerfassung *(Forts.)*
planen 109
terminieren 109
verzögert 109
Datenerfassung terminieren 109
Datenerfassungs-Hierarchie 23
Datenerfassungs-Routes zurücksetzen 23, 96
Datenerfassungsmanager 96
Datenloser Endpunktmodus 166
Datenspeicherung im Konfigurations-Repository 206
DB2
Exemplar erstellen 33
Konfiguration 32
Konfigurations-Repository erstellen 64, 69
RIM-Host 32
DEBUG_EXT_PRESENT, Beschreibung 400
Depot
Definition 4
Fehlerbehebung 530
Größe erhöhen 250
Inhalt überprüfen 527
vergrößern 97
verschieben 96
DEV_ADDR, Beschreibung 410
DEV_CLASS, Beschreibung 410
DEV_IS_HUB, Beschreibung 411
DEV_NAME, Beschreibung 395
DEV_SUBCLASS, Beschreibung 410
Dialog
Abfrage ausführen 222
Abfrage erstellen 211, 213
Abfrageergebnisse exportieren 222
Administrator erstellen 131
Anmeldenamen festlegen 132
Datei-Browser 26
Geplanten Job hinzufügen 203
Inventory-Profil 200
Konfiguration für DMI-Suchfunktion 186
Konfiguration für Softwaresuchlauf 179
Profil erstellen 169
Profil klonen 192, 193
Profile löschen 195
Programmkorrektur installieren 25, 27, 28, 29
Spalte 219
Dialog *(Forts.)*
Tivoli Hardware-Suchfunktion 185, 189
TMR-Berechtigungsklassen 131
Zeitlimiteinstellungen 201
Dispatcher-Nummer für Objekte
inaktive Verbindungen angeben 111
DMI
dmi_RDBMS_type _schema.sql, Skript 187
dmi_RDBMS_type _schema.sql script 515
Fehlerbehebung 511, 515
Fehlernachrichten 512
Suchfunktion 185
Suchläufe anpassen 185
Tabellen und Spalten benennen 515
dmi_RDBMS_type _schema.sql 187, 515
dmiscan 514
DOMAIN_NAME, Beschreibung 397
DRV_NAME, Beschreibung 403
DRV_VERS, Beschreibung 403

E

Editieren
Benutzerdatenformular 144, 152
Konfigurations-Repository 135, 158
Softwaresignaturen 122, 126
useradd.mif, Datei 144, 152
Einsatzbereich von Tivoli Inventory 2
Endpunkt-initiierte Suchläufe
CLI-Informationen 271
Einstellungen für Umgebung 271
erforderliche Tasks 517
Fehlerbehebung 515, 517
Suchlaufdaten in eine Protokolldatei schreiben 272
Endpunkte
Definition 7
Suchlaufdaten löschen 311
Endpunktinitiierte Suchläufe
Beschreibung 204
Erfassung
beendete 5
Fehlerbedingungen 5
planen 113

Erfassung *(Forts.)*
verzögert 113
verzögerte 5
Ergebnisse, Seite 150
Ergebnisseite 147
Erstellen
Abfragen 211, 220
Benutzerdatenformular 144, 152
Callback-Objekt von Inventory 256
Inventory Data Handler 258
Konfigurations-Repositorys 63, 90
Profile 165, 171
Protokolltabellen für benutzerspezifische Tabellen erstellen 140
Protokolltabellen für DB2 68
Protokolltabellen für Informix 71
Protokolltabellen für MS SQL-Server 75
Protokolltabellen für Oracle 80
Protokolltabellen für Sybase 84
RIM-Objekte 105
Softwaresignaturen 122, 126
useradd.mif, Datei 144, 152
EXE-Dateien, Suche nach 182, 343, 359

F

FAST_FLOAT_SAVE, Beschreibung 400
FAST_SYS_CALL, Beschreibung 400
Fehlerbedingungen 5, 512, 522
Fehlerbehebung
Abfragen 535
DMI 511, 515
Endpunkt-initiierte Suchläufe 515, 517
GUI 518, 520
Konfigurations-Repository 511
SCS 534
SCS (Scalable Collection Service) 520
Fenster
Hardware 183, 188
Profilmanager 168, 170
Signaturen 124
Skripts und MIF-Dateien 190
Software 174, 177
Fenster 'Hardware' 183, 188
Fettschrift, Verwendung xxi
FILE_DESC, Tabelle 441
FILE_FILTER, Tabelle 442
FILE_GROUP, Beschreibung 377
FILE_NAME (installierte Datei), Beschreibung 377
FILE_NAME (nicht abgeglichen), Beschreibung 389
FILE_OWNER, Beschreibung 377
FILE_PATH, Beschreibung 387
FILE_PATH, Tabelle 442
FILE_PERMISSIONS, Beschreibung 377
FILE_SIZE (installierte Datei), Beschreibung 377
FILE_SIZE (nicht abgeglichen), Beschreibung 389
Filter
nur nach .exe-Dateien suchen 182
nur nach EXE-Dateien suchen 343, 359
Suche nach grundlegenden Dateiinformationen 181, 308, 342, 358
FLPY_DRV_QUERY 416
FLPY_DRV_VIEW 374
Fremdschlüssel 137, 142
FS_ACCESS_POINT, Beschreibung 395
FS_FREE_SIZE_KB, Beschreibung 396
FS_MOUNT_POINT, Beschreibung 396
FS_TOTAL_SIZE_KB, Beschreibung 396
FS_TYPE, Beschreibung 396
FUNCTION_KEYS, Beschreibung 384

G

Gateway, Definition 7
Gemeinsamer Einsatz von Anwendungen 16
Geplanten Job hinzufügen, Dialog 203
Größe der Dateneinheiten 98, 113
Größe von Dateneinheiten 250
Grundlegende Dateiinformationen
filtern 181, 308, 342, 358
suchen 181, 342, 358
GUI
Fehlerbehebung 518, 520
Protokolldatei aktivieren 518

GUI *(Forts.)*
Systemvoraussetzungen 518
UNIX-Verbindungen aktivieren 519

H

H_AIX_SUBSCRIPTIONS 438
H_CDROM_QUERY 435
H_CDROM_VIEW 414
H_COMPUTER, Tabelle 460
H_COMPUTER_MEM_QUERY 435
H_COMPUTER_QUERY 435
H_COMPUTER_SYS_MEM, Tabelle 460
H_COMPUTER_VIEW 414
H_FLPY_DRV_QUERY 435
H_FLPY_DRV_VIEW 414
H_HDISK_QUERY 435
H_HDISK_VIEW 414
H_HEADER_INFO_QUERY 435
H_HEADER_VIEW 414
H_HPUX_SUBSCRIPTIONS 438
H_INST_FILE_QUERY 435
H_INST_FILE_VIEW 414
H_INST_HEADER_INFO, Tabelle 460
H_INST_MOUSE, Tabelle 460
H_INST_NATIVE_SWARE, Tabelle 460
H_INST_PARTITION, Tabelle 460
H_INST_PRINTER, Tabelle 460
H_INST_PROCESSOR, Tabelle 460
H_INST_SWARE_VIEW 414
H_INST_USB_DEV, Tabelle 460
H_INST_VID_CARD, Tabelle 461
h_inv_db2_schema.log 68
h_inv_db2_schema.sql, Skript 68
h_inv_infx_schema.log 71
h_inv_infx_schema.sql, Skript 71
h_inv_ms_schema.log 75
h_inv_ms_schema.sql, Skript 75
h_inv_ora_schema.log 80
h_inv_ora_schema.sql, Skript 80
h_inv_syb_schema.log 84
h_inv_syb_schema.sql, Skript 84
h_inventory_query.sh, Skript 87
H_INVENTORY_SWARE 435
H_IP_ADDR, Tabelle 461
H_IP_ADDR_QUERY 435
H_IP_ADDR_VIEW 414
H_IPX_ADDR, Tabelle 461
H_IPX_ADDR_QUERY 435
H_IPX_ADDR_VIEW 414
H_MATCHED_SWARE, Tabelle 461
H_MEM_VIEW 414
H_MOUSE_QUERY 435
H_MOUSE_VIEW 414
H_NATIV_SWARE_QUERY 435
H_NATIV_VIEW 414
H_NET_ADAPTER, Tabelle 461
H_NET_CARD_QUERY 435
H_NET_CARD_VIEW 414
H_NETWARE_SUBSCRIPTIONS 438
H_NOSIG_FILES_QUERY 436
H_NOSIG_FILES_VIEW 414
H_NW_SERVER, Tabelle 461
H_NW_SERVER_QUERY 436
H_NW_SERVER_VIEW 414
H_NW_VOLS, Tabelle 461
H_NW_VOLS_QUERY 436
H_NW_VOLS_VIEW 414
H_OS_QUERY 436
H_OS_VIEW 414
H_OS2_SUBSCRIPTIONS 438
H_PARTITION_QUERY 436
H_PARTITION_VIEW 414
H_PC_BIOS_QUERY 436
H_PC_BIOS_VIEW 414
H_PC_SYS_PARAMS, Tabelle 461
H_PCI_DEV, Tabelle 461
H_PCI_DEV_QUERY 436
H_PCI_DEV_VIEW 415
H_PCPROCESSOR_QUERY 436
H_PCPROCESSOR_VIEW 415
H_PRINTER_QUERY 436
H_PRINTER_VIEW 415
H_PROCESSOR_QUERY 436
H_PROCESSOR_VIEW 415
H_Solaris_SUBSCRIPTIONS 438
H_STORAGE_DEV, Tabelle 461
H_STORAGE_DEV_QUERY 436
H_STORAGE_DEV_VIEW 415
h_subscription_query.sh 87

H_TAPEDRV_QUERY 436
H_TAPEDRV_VIEW 415
H_UNIX_SYS_PARAMS, Tabelle 461
H_UNIX_SYS_QUERY 436
H_UNIX_SYS_VIEW 415
H_UNMATCHED_FILES, Tabelle 461
H_USB_DEV_QUERY 436
H_USB_DEV_VIEW 415
H_VID_CARD_QUERY 436
H_VID_CARD_VIEW 415
H_WIN_2000_SUBSCRIPTIONS 438
H_WIN_98_SUBSCRIPTIONS 439
H_WIN_ALL_SUBSCRIPTIONS 439
H_WIN_ME_SUBSCRIPTIONS 439
H_WIN_NT_SUBSCRIPTIONS 439
HDISK, Tabelle 443
HDISK_CYLINDERS, Beschreibung 375
HDISK_HEADS, Beschreibung 375
HDISK_QUERY 417
HDISK_SECTORS, Beschreibung 375
HDISK_SIZE_KB, Beschreibung 375
HDISK_VIEW 375
HEADER_INFO, Tabelle 443
HEADER_INFO_QUERY 418
HEADER_INFO_VIEW 376
Header-Informationen, Suchen 180, 342
HEADER_NAME, Beschreibung 376
HEADER_PUBLISHER, Beschreibung 376
HEADER_VERS, Beschreibung 376
Hierarchie
  Kollektor 4, 5, 22, 95, 96
  Repeater 4, 22, 95
Hinweisgruppe 198
HOST_CNTRL, Beschreibung 410
HPUX_SUBSCRIPTIONS 437

I

idlcall 108
inaktive Verbindung 109
Inaktive Verbindung 112, 530
Informix
  Konfiguration 35
  Konfigurations-Repository erstellen 71
Informix *(Forts.)*
  RIM-Host 35
Informix-Datenbank als
  Konfigurations-Repository erstellen 69
INST_DATE, Beschreibung 388
INST_FILE_QUERY 418
INST_FILE_VIEW 377
INST_HEADER_INFO, Tabelle 444
INST_MOUSE, Tabelle 444
INST_NATIV_SWARE, Tabelle 445
INST_PARTITION, Tabelle 445
INST_PCI_ID, Beschreibung 402
INST_PRINTER, Tabelle 446
INST_PROCESSOR, Tabelle 446
INST_SWARE_VIEW 379
INST_USB_DEV, Tabelle 447
INST_VID_CARD, Tabelle 447
Install_client, Berechtigungsklasse 236
Install_product, Berechtigungsklasse 236
Installieren
  Abfragen 87
  SCS über die Arbeitsoberfläche 25
  SCS über die Befehlszeile 29
  Softwaresignaturen 85
  Subskriptionsabfragen 87
  Tivoli Inventory
    mit Software Installation Service 21
    Übersicht 19
  Tivoli Inventory Gateway 55
  Tivoli Inventory Server
    auf dem TMR-Server 43, 54
    auf verwalteten Knoten 43, 54
  zwischen Tivoli Inventory Server oder Gateway auswählen 43
Integrierende Anwendungen 16
inv_40 51
inv_db2_admin.sql, Skript 65
inv_db2_schema.sql, Skript 67
inv_infx_schema.sql 71
inv_ms_sql_admin.sql, Skript 73
inv_ms_sql_schema.sql, Skript 75
inv_ora_admin.sql, Skript 77
inv_ora_schema.sql, Skript 79
inv_query 51
INV_SA.DAT 273
inv_scan_nn.log 524

inv_syb_admin.sql, Skript 82
inv_syb_schema.sql, Skript 84
invdh_1 51, 52, 105
Inventory Data Handler
  Ausgabethreads konfigurieren 101
  Definition 5
  erstellen 258
  Host auswählen 45
  Konfiguration von Information 117
  Konfigurationsdaten anzeigen 116, 274
  konfigurieren 100, 258, 321
  Statusinformationen konfigurieren 100, 259, 321
  verschieben 102
  verwenden 99
  Wiederholungen konfigurieren 102, 260, 323
  Zeitlimit zwischen Wiederholungen konfigurieren 102, 261, 323
Inventory-Daten anzeigen
  Prozedur 229, 232
  wqueryinv 318
  zu einem Client 229
Inventory_end_user, Berechtigungsklasse 235
INVENTORY_HWARE 419
Inventory-Profil, Dialog 200
Inventory-Profil, Dialogfenster 200
Inventory-Profile 12
Inventory_query, Berechtigungsklasse 235
inventory_query.sh, Skript 87
Inventory-RIM-Objekt 51
Inventory_scan, Berechtigungsklasse 235
INVENTORY_SWARE 420
Inventory_view, Berechtigungsklasse 235
InventoryConfig
  einem Richtlinienbereich hinzufügen 167, 169
  einem Richtlinienbereich zuordnen 50
  Standardwerte festlegen 463
INVENTORYDATA 380
IP_ADDR, Beschreibung 381
IP_ADDR, Tabelle 448
IP_ADDR_QUERY 420
IP_ADDR_VIEW 381
IP_DOMAIN, Beschreibung 382
IP_GATEWAY, Beschreibung 382
IP_HOSTNAME, Beschreibung 382
IP_PRIMARY_DNS, Beschreibung 382
IP_SECONDARY_DNS, Beschreibung 382
IP_SUBNET, Beschreibung 382
IPX_ADDR, Beschreibung 383
IPX_ADDR, Tabelle 449
IPX_ADDR_QUERY 421
IPX_ADDR_VIEW 383

K

KEYBOARD_QUERY 421
KEYBOARD_TYPE, Beschreibung 384
KEYBOARD_VIEW 384
Klonen von Profilen 192, 194
Kollektor
  das Depot vergrößern 97, 530
  das Depot verschieben 96
  Datenerfassung terminieren 109
  Datenerfassungs-Routes zurücksetzen 23, 96
  Definition 4
  Depotgröße erhöhen 250
  Fehlerbehebung 530
  Kollektor-Routes zurücksetzen 252
  Konfigurationsdaten anzeigen 115, 248
  Laufzeitverzeichnis verschieben 96, 249
  Protokolldatei konfigurieren 249, 253, 522
  Sammlungen planen 252
  starten 96, 248
  Statusinformationen anzeigen 115, 262
  stoppen 96, 248
  Threads konfigurieren 98, 113, 250, 251
  verwenden 95
  Wiederholungen konfigurieren 98, 251
  zurücksetzen 254
Kollektor-Manager 5, 252, 523
Kollektor-Routes zurücksetzen 252
Kollektor starten 248
Kollektor stoppen 248
Kollektor zurücksetzen 254
Kollektoren starten 96
Kollektoren stoppen 96
Kollektorhierarchie 4, 5, 22, 95, 96
Komponenten von Tivoli Inventory 3

Konfiguration für DMI-Suchfunktion, Dialogfenster 186
Konfiguration für Softwaresuchlauf, Dialogfenster 179
Konfiguration prüfen 68, 71, 76, 80, 84
Konfigurations-Repository
  anpassen 158
  Anpassen 135
  Ansichten 370, 415
  Anzeigen hinzufügen 156
  Daten für einen Endpunkt löschen 311
  Daten speichern 206
  Definition 3
  Editieren 135, 158
  Fehlerbehebung 511
  Filterliste ändern 181, 308
  für RDBMS erstellen
    DB2 32, 65, 69
    Informix 35, 70, 72
    MS SQL-Server 72, 76
    Oracle 77, 81
    Sybase 81, 85
    Übersicht 63
  Inventory-Daten anzeigen 117
  mehrstufige Löschoperation 142
  Protokollansichten 413, 415
  Protokollsubskriptionsabfragen 438, 439
  Schemata 67, 71, 75, 79, 84
  Softwaresignaturen ändern 122, 313
  Standardabfragen 415, 439
  Subskriptionsabfragen 436, 439
  Suchläufe für Senden von Daten konfigurieren 175, 343, 358
  Tabellen
    standardmäßig in Tivoli Inventory 439, 461
  Tabellen hinzufügen 141
  Übersicht 439, 461
Konfigurationsdateien
  Inhalt in eine Protokolldatei schreiben 272
  Richtlinie 476
  verteilen 173, 326
Kontrollsummen generieren 181, 344, 358
Kopieren von Profilen 192
Kursivschrift, Verwendung xxi

L

Laufzeitverzeichnis
  Standardposition 249
  Standardverzeichnis 97
  verschieben 96, 249
Leistungsmerkmale von Tivoli Inventory 12
LINK_SPEED, Beschreibung 383
Löschen
  Profile 194, 196
  Softwaresignaturen 122, 126
  Suchlaufdaten für einen Endpunkt 311

M

MACHINECHECK_ARCH, Beschreibung 399
MACHINECHECK_EXCPT, Beschreibung 399
Management Information Format (MIF) 7
MANUFACTURER (CD-ROM), Beschreibung 371
MANUFACTURER (Diskette), Beschreibung 374
MANUFACTURER (Festplatte), Beschreibung 375
MANUFACTURER (Netzwerkadapter), Beschreibung 388
MANUFACTURER (PC-Prozessor), Beschreibung 398
MANUFACTURER (Prozessor), Beschreibung 404
MANUFACTURER (Speichergerät), Beschreibung 405
MANUFACTURER (USB), Beschreibung 411
MATCH_CRC32_QUERY 422
MATCH_MD5_QUERY 422
MATCH_QUICK_QUERY 423
MATCH_SWARE_VIEW 385
MATCHED_SWARE, Tabelle 449
MAX_PACKET_SIZE, Beschreibung 383
mcollect.log 249, 522, 523
MD5 344
MDist 2 4, 22, 199
MEDIA_TYPE, Beschreibung 395
mehrstufige Löschoperation 142

MEM_TYPE_RANGE_REG, Beschreibung 399
MIF-Attribut hinzufügen, Seite 149
MIF-Attribute hinzufügen bzw. löschen, Seite 148
MIF-Benutzer
  benutzerspezifische
    mit Tivoli Inventory verwenden 162
  Dateien
    benennen 159
MIF-Benutzerdateien
  benennen 137, 159
  Richtlinie 484
  verwenden 158, 162
MIF-Dateien
  Beispiel 160
  benutzerdefiniert
    Richtlinie 484
  benutzerdefinierte
    mit Tivoli Inventory verwenden 189, 190, 331, 348
  benutzerspezifische
    mit Tivoli Inventory verwenden 158
  Definition 7
MIF-Dateien,
  die von Tivoli Inventory erstellt werden 160
MIF-Gruppe hinzufügen, Seite 146
MIF-Gruppen und -Attribute, Seite 150
MMX_TECHNOLOGY, Beschreibung 400
Mobile Endpunkte
  Erinnerungsnachrichten 267
  Unterstützung 14, 268
  verbindliche Verteilungen 266
  Verteilungen ausblenden 268, 327
MODEL (CD-ROM), Beschreibung 371
MODEL (Diskette), Beschreibung 374
MODEL (Festplatte), Beschreibung 375
MODEL (Speichergerät), Beschreibung 405
MODEL_SPECIFIC_REG, Beschreibung 399
MODIFIED_TIME, Beschreibung 377
Monospace-Schrift, Verwendung xxi
MOUSE, Tabelle 449
MOUSE_MODEL, Beschreibung 386
MOUSE_QUERY 424
MOUSE_TYPE, Beschreibung 386
MOUSE_VIEW 386
mrmbios 162
MS SQL-Server
  Konfiguration 35
  Konfigurations-Repository erstellen 72, 76
  RIM-Host 36
Multiplexverteilungs-Service 4

N

Namensregistrierdatenbank 119, 242
NATIV_SWARE, Tabelle 450
NATIV_SWARE_QUERY 424
NATIV_SWARE_VIEW 387
NET_ADAPTER, Tabelle 451
NET_CARD_QUERY 425
NET_CARD_VIEW 388
NET_NUM, Beschreibung 383
NETWARE_SUBSCRIPTIONS 437
nobody, Benutzereintrag 249
NODE_ADDR, Beschreibung 383
NOSIG_FILES_QUERY 425
NOSIG_FILES_VIEW 389
NOW_3_D_ARCH, Beschreibung 401
NUM_OF_PORTS, Beschreibung 411
Nummer der Region 233
NW_ACCOUNTING_VERS, Beschreibung 391
NW_CLIB_MAJOR_VERS, Beschreibung 392
NW_CLIB_MINOR_VERS, Beschreibung 392
NW_CLIB_REVISION, Beschreibung 392
NW_DEV_NAME, Beschreibung 390
NW_INET_BRG_SUPP, Beschreibung 392
NW_MAX_CONNS, Beschreibung 390
NW_MAX_CONNS_USED, Beschreibung 391
NW_MAX_VOLS, Beschreibung 390
NW_PRINTSERVR_VERS, Beschreibung 391
NW_QUEING_VERS, Beschreibung 391
NW_REVISION_LEVEL, Beschreibung 390
NW_SER_NUM, Beschreibung 392
NW_SERVER, Tabelle 451
NW_SERVER_QUERY 426
NW_SERVER_VIEW 390
NW_SFT_LEVEL, Beschreibung 391
NW_SUB_VERS, Beschreibung 390
NW_TTS_LEVEL, Beschreibung 391
NW_VAP_VERS, Beschreibung 391

NW_VERS, Beschreibung 390
NW_VIRT_CONS, Beschreibung 391
NW_VOLS_QUERY 427
NW_VOLS_VIEW 393
NWVOL_AVAIL_BLKS, Beschreibung 393
NWVOL_AVAIL_SLOTS, Beschreibung 393
NWVOL_BLK_SECTORS, Beschreibung 393
NWVOL_DIR_SLOTS, Beschreibung 393
NWVOL_IS_REMOVABLE, Beschreibung 394
NWVOL_NAME, Beschreibung 393
NWVOL_TOTAL_BLKS, Beschreibung 393

O

Objekt-Dispatcher-Nummer
  Offlinks angeben 252
Objekt-ID 233
Objekt-Nummer 233
Objektpfade 242
Objektverweise 241
odadmin 23, 111, 520, 534
Offlink 252
ON_CHIP_APIC, Beschreibung 399
ON_CHIP_FPU, Beschreibung 400
Oracle
  Konfiguration 36
  Konfigurations-Repository erstellen 77, 81
  RIM-Host 36
OS_INST_DATE, Beschreibung 395
OS_MAJOR_VERS, Beschreibung 394
OS_MINOR_VERS, Beschreibung 395
OS_NAME, Beschreibung 394
OS_QUERY 427
OS_SUB_VERS, Beschreibung 395
OS_TYPE, Beschreibung 394
OS_VIEW 394
OS2_SUBSCRIPTIONS 437

P

PACKAGE_ID, Beschreibung 387
PACKAGE_NAME, Beschreibung 387
PACKAGE_VERS, Beschreibung 387
PAGE_ATTR_TABLE, Beschreibung 400
PAGE_GLOBAL_ENABLE, Beschreibung 399
PAGE_SIZE, Beschreibung 372
PAGE_SIZE_EXT, Beschreibung 398
PAGE_SIZE_EXT36, Beschreibung 400
PARENT_ADDR, Beschreibung 410
PARTITION_QUERY 428
PARTITION_TYPE, Beschreibung 395
PARTITION_VIEW 395
PATH (installierte Datei), Beschreibung 377
PATH (nicht abgeglichen), Beschreibung 389, 391
PC_BIOS_QUERY 428
PC_BIOS_VIEW 397
PC-Hardwaresuchläufe 183, 337
PC_PROCESSOR_QUERY 429
PC_PROCESSOR_VIEW 398
PC_SYS_PARAMS, Tabelle 453
PCI_DEV, Tabelle 454
PCI_DEV_NAME, Beschreibung 402
PCI_DEV_QUERY 430
PCI_DEV_VIEW 402
PERM_MAC_ADDR, Beschreibung 388
Pfad
  absolut 242
  relativ 242
PHYSICAL_ADDR_EXT, Beschreibung 399
PHYSICAL_FREE_KB, Beschreibung 372
PHYSICAL_SIZE_KB, Beschreibung 396
PHYSICAL_TOTAL_KB, Beschreibung 372
PORT_NAME, Beschreibung 403
PORT_NUM, Beschreibung 410
PRFL_ACTION, Beschreibung 413
Primärschlüssel 137, 159
PRINTER, Tabelle 454
PRINTER_IS_LOCAL, Beschreibung 403
PRINTER_LOCATION, Beschreibung 403
PRINTER_MODEL, Beschreibung 403
PRINTER_NAME, Beschreibung 403
PRINTER_QUERY 431
PRINTER_VIEW 403

PROCESSOR, Tabelle 454
PROCESSOR_MODEL (PC-Prozessor), Beschreibung 398
PROCESSOR_MODEL (Prozessor), Beschreibung 404
PROCESSOR_QUERY 431
PROCESSOR_SPEED (PC-Prozessor), Beschreibung 398
PROCESSOR_SPEED (Prozessor), Beschreibung 404
PROCESSOR_VIEW 404
PRODUCT, Beschreibung 411
PRODUCT_ID, Beschreibung 411
Profil erstellen, Dialog 169
Profil klonen, Dialogfenster 192, 193
Profile
  abzusuchende Dateien festlegen 331, 348
  Abzusuchende Dateien festlegen 182
  abzusuchende Verzeichnisse festlegen 331, 348
  Abzusuchende Verzeichnisse festlegen 183
  allgemeine Merkmale anzeigen 277
  allgemeine Merkmale definieren 172, 325
  anpassen 171, 191
  Anpassen 12
  auszuführende benutzerdefinierte Skripts festlegen 189, 331, 348
  Benachrichtigungsintervall einstellen 202, 266
  benennen 166
  benutzerdefinierte MIF-Dateien abrufen 189, 331, 348
  Daten an Konfigurations-Repository senden 175, 343, 358
  Datenoptionen festlegen 173, 329
  Definition 6
  Erinnerungsnachrichten an Endpunkte für standortunabhängigen Zugriff 267
  erstellen 165, 171
  Fehlerbehebung 529
  globale Merkmale anzeigen 117
  Informationen anzeigen 277
  Informationen anzeigen zu 117
  klonen 192, 194
  Konfigurationsdaten anzeigen 282, 294
  kopieren 192
Profile *(Forts.)*
  löschen 194, 196
  Optionen für Hardwaresuchläufe anzeigen 285, 297
  Optionen für Hardwaresuchläufe festlegen 183, 189, 337, 354
  Optionen für Softwaresuchläufe anzeigen 288, 300
  Optionen für Softwaresuchläufe festlegen 174, 183, 342, 358
  Prioritäten einstellen 199, 265
  Prioritäten festlegen 200
  Standardwerte für neue Profile festlegen 464
  Statusoptionen festlegen 117, 119, 326
  Termin für Verteilung festlegen 201, 266
  Übersicht über die Verteilung 198
  umbenennen 196
  Unterstützung von Endpunkten für standortunabhängigen Zugriff 268
  verbindliche Verteilungen an Endpunkte für standortunabhängigen Zugriff 266
  verteilen 200, 204, 265
  Verteilungen an Endpunkte für standortunabhängigen Zugriff ausblenden 268, 327
  Verteilungsoptionen festlegen 173, 326
  verwalten 165
  verwenden 164
  wiederherstellen 195
  WoL-Unterstützung 269, 329
  Zeitlimit für Endpunktantwort festlegen 202, 266
  Zeitlimit für Profilverteilung festlegen 202, 266
  Zeitlimit für Suchlaufversuche festlegen 326
Profile löschen, Dialog 195
Profilmanager
  Definition 6
  verwenden 165
Profilmanager - Fenster 168, 170
Profilverteilung
  CLI-Informationen 265
  Prozedur 204
  Übersicht 198
  Verfahren 200
Programmkorrektur installieren, Dialog 25, 27, 28, 29

Protokolldatei DEBUG 3 519
Protokollfunktion
aktivieren 367
Protokolltabellen verwenden 367
Übersicht 366
Protokolltabellen
für benutzerspezifische Tabellen erstellen 140
für DB2 erstellen 68
für Informix erstellen 71
für MS SQL-Server erstellen 75
für Oracle erstellen 80
für Sybase erstellen 84
verwenden 367
Prozeduren
benutzerdefinierte 331, 348
dmi_RDBMS_type _schema.sql 515
PUBLISHER, Beschreibung 387

Q

Query_edit, Berechtigungsklasse 235
Query_execute, Berechtigungsklasse 235
Query_view, Berechtigungsklasse 235
QUERY_VIEWS, Tabelle 157

R

RDBMS
Konfiguration 31
RDBMS-
Server 31
RDBMS Interface Module (RIM) 3
RECORD_ACTION, Beschreibung 413
RECORD_TIME, Beschreibung 371
Referenzliteratur xvii
Registrierter Name, Definition 241
Relativer Pfad 242
Release-Notes xvii
Repeater-Hierarchie 4, 22, 95
Repeater-Stationen 4
Ressourcenart, InventoryConfig 169
Ressourcenart 'InventoryConfig' 50
restore, Berechtigungsklasse 236
Richtlinie
einem Richtlinienbereich zuweisen 469
ersetzen 467
erstellen 466
ic_def_global_action 475
ic_def_global_ep_timeout 477
ic_def_global_logfile_host 478
ic_def_global_logfile_path 479
ic_def_global_notice_interval 480
ic_def_global_notice_location 481
ic_def_global_notice_type 482
ic_def_global_update_replace 483
ic_def_pc_custom_mifs 484
ic_def_pc_custom_script 485
ic_def_pc_exclude_dirs 486
ic_def_pc_extensions 487
ic_def_pc_hw_gran 488
ic_def_pc_hw_outfile 490
ic_def_pc_hw_scan 491
ic_def_pc_include_dirs 492
ic_def_pc_sw_crc 493
ic_def_pc_sw_flags 494
ic_def_pc_sw_outfile 495
ic_def_pc_sw_scan 497
ic_def_unix_custom_mifs 498
ic_def_unix_custom_script 499
ic_def_unix_exclude_dirs 500
ic_def_unix_extensions 501
ic_def_unix_hw_gran 502
ic_def_unix_hw_outfile 503
ic_def_unix_hw_scan 504
ic_def_unix_include_dirs 505
ic_def_unix_sw_crc 506
ic_def_unix_sw_flags 507
ic_def_unix_sw_outfile 508
ic_def_unix_sw_scan 510
Standardmethode festlegen 470
Standardwert 464
Tivoli Inventory 463, 510
Übersicht 463
Richtlinie zuweisen 469
rim_db_log 526

RIM-Host
auswählen 30, 46
Definition 3
mit DB2 32
mit Informix 35
mit MS SQL-Server 36
mit Oracle 36
mit Sybase 37
siehe auch 'RIM-Objekt' 105
TMR-Verbund 32
RIM-Objekt
Anwendungskennung angeben 105
Definition 6
erstellen 105
inv_40 51
inv_query 51
invdh_1 51, 52, 105
Inventory 51
konfigurieren 105
maximale Anzahl der Datenbankverbindungen angeben 105
siehe auch 'RIM-Host' 105
RIM_update, Berechtigungsklasse 145, 235
RIM_view, Berechtigungsklasse 145, 235

S

sa_config.log 272, 517
sa_results.log 272, 517
Sammlung
planen 252
verzögert 252
Sammlungen planen 252
Scheduler-Dämon 5
Schnittstellendatei 37
Schriftbilder, Verwendung xxi
SCS
Datenfluss steuern 112
erforderliche Komponenten 4
Fehlerbehebung 534
Funktionen 15
Informationen abrufen zu 114
über die Arbeitsoberfläche installieren 25
über die Befehlszeile installieren 29
SCS *(Forts.)*
zusammen mit Tivoli Inventory verwenden 8, 93
SCS (Scalable Collection Service)
Befehle 245
Fehlerbehebung 520
Seite
Benutzerdatenformular 152, 154
Benutzerdatenschablone 146
Ergebnis 147
Ergebnisse 150
MIF-Attribut hinzufügen 149
MIF-Attribut hinzufügen bzw. löschen 148
MIF-Gruppe hinzufügen 146
MIF-Gruppen und -Attribute 150
UserLink for Tivoli 153
UserLink for Tivoli Inventory 154
senior, Berechtigungsklasse 236
SER_NUM (CD-ROM), Beschreibung 371
SER_NUM (Festplatte), Beschreibung 375
SER_NUM (PC-Prozessor), Beschreibung 401
SER_NUM (Prozessor), Beschreibung 404
SER_NUM (Speichergerät), Beschreibung 405
SER_NUM (USB), Beschreibung 410
SER_NUM_ENABLED, Beschreibung 400
Sicherheit 17
SIMD_EXT_SUPP, Beschreibung 400
Skalierbare Erfassung 8
Skalierbarer Erfassungsservice (SCS = Scalable Collection Service) 4
Skripts
benutzerdefinierte 189, 190
dmi_RDBMS_type _schema.sql 187
h_inv_db2_schema.sql 68
h_inv_infx_schema.sql 71
h_inv_ms_schema.sql 75
h_inv_ora_schema.sql 80
h_inv_syb_schema.sql 84
h_inventory_query.sh 87
h_subscription_query.sh 87
inv_db2_admin.sql 65
inv_db2_schema.sql 67
inv_infx_schema.sql 71
inv_ms_sql_admin.sql 73
inv_ms_sql_schema.sql 75
inv_ora_admin.sql 77

Skripts *(Forts.)*
inv_ora_schema.sql 79
inv_syb_admin.sql 82
inv_syb_schema.sql 84
inventory_query.sh 87
Richtlinie 485
subscription_query.sh 87
Skripts und MIF-Dateien, Fenster 190
Software, Fenster 174, 177
Software Installation Service (SIS) 21
Softwaresignaturen
Definition 122
editieren 122, 126, 313
erstellen 122, 126, 313
installieren 85
löschen 122, 126, 313
suchen über 179, 342, 358
Solaris_SUBSCRIPTIONS 437
Spalte, Dialog 219
Spalten
benennen 136
für DB2- und Informix-Datenbanksysteme 136, 138
Spaltenname, Beschreibung
ACCESSED_TIME 377
ADAPTER_MODEL 388
ADAPTER_TYPE 388
BIOS_DATE 397
BIOS_ID 397
BIOS_ID_BYTES 397
BIOS_MANUFACTURER 397
BIOS_MODEL 397
BIOS_SER_NUM 397
BIOS_STRING 397
BUTTONS 386
CHECKSUM_CRC32 378
CHECKSUM_MD5 378
CHECKSUM_QUICK 378
CHIP_FAMILY 398
CHIP_MODEL 398
CHIP_STEPPING 398
CMPXCHG8B_SUPP 399
COMPUTER_MODEL 373
COMPUTER_SCANTIME 373
COMPUTER_SYS_ID 371, 372
COND_MOVE_SUPP 399
Spaltenname, Beschreibung *(Forts.)*
CREATED_TIME 377
CURRENT_ADDR 388
DEBUG_EXT_PRESENT 400
DEV_ADDR 410
DEV_CLASS 410
DEV_IS_HUB 411
DEV_NAME 395
DEV_SUBCLASS 410
DOMAIN_NAME 397
DRV_NAME 403
DRV_VERS 403
FAST_FLOAT_SAVE 400
FAST_SYS_CALL 400
FILE_GROUP 377
FILE_NAME (installierte Datei) 377
FILE_NAME (nicht abgeglichen) 389
FILE_OWNER 377
FILE_PATH 387
FILE_PERMISSIONS 377
FILE_SIZE (installierte Datei) 377
FILE_SIZE (nicht abgeglichen) 389
FS_ACCESS_POINT 395
FS_FREE_SIZE_KB 396
FS_MOUNT_POINT 396
FS_TOTAL_SIZE_KB 396
FS_TYPE 396
FUNCTION_KEYS 384
HDISK_CYLINDERS 375
HDISK_HEADS 375
HDISK_SECTORS 375
HDISK_SIZE_KB 375
HEADER_NAME 376
HEADER_PUBLISHER 376
HEADER_VERS 376
HOST_CNTRL 410
INST_DATE 388
INST_PCI_ID 402
IP_ADDR 381
IP_DOMAIN 382
IP_GATEWAY 382
IP_HOSTNAME 382
IP_PRIMARY_DNS 382
IP_SECONDARY_DNS 382
IP_SUBNET 382
IPX_ADDR 383

Spaltenname, Beschreibung *(Forts.)*
KEYBOARD_TYPE 384
LINK_SPEED 383
MACHINECHECK_ARCH 399
MACHINECHECK_EXCPT 399
MANUFACTURER (CD-ROM) 371
MANUFACTURER (Diskette) 374
MANUFACTURER (Festplatte) 375
MANUFACTURER (Netzwerkadapter) 388
MANUFACTURER (PC-Prozessor) 398
MANUFACTURER (Prozessor) 404
MANUFACTURER (Speichergerät) 405
MANUFACTURER (USB) 411
MAX_PACKET_SIZE 383
MEDIA_TYPE 395
MEM_TYPE_RANGE_REG 399
MMX_TECHNOLOGY 400
MODEL (CD-ROM) 371
MODEL (Diskette) 374
MODEL (Festplatte) 375
MODEL (Speichergerät) 405
MODEL_SPECIFIC_REG 399
MODIFIED_TIME 377
MOUSE_MODEL 386
MOUSE_TYPE 386
NET_NUM 383
NODE_ADDR 383
NOW_3_D_ARCH 401
NUM_OF_PORTS 411
NW_ACCOUNTING_VERS 391
NW_CLIB_MAJOR_VERS 392
NW_CLIB_MINOR_VERS 392
NW_CLIB_REVISION 392
NW_DEV_NAME 390
NW_INET_BRG_SUPP 392
NW_MAX_CONNS 390
NW_MAX_CONNS_USED 391
NW_MAX_VOLS 390
NW_PRINTSERVR_VERS 391
NW_QUEING_VERS 391
NW_REVISION_LEVEL 390
NW_SER_NUM 392
NW_SFT_LEVEL 391
NW_SUB_VERS 390
NW_TTS_LEVEL 391
NW_VAP_VERS 391

Spaltenname, Beschreibung *(Forts.)*
NW_VERS 390
NW_VIRT_CONS 391
NWVOL_AVAIL_BLKS 393
NWVOL_AVAIL_SLOTS 393
NWVOL_BLK_SECTORS 393
NWVOL_DIR_SLOTS 393
NWVOL_IS_REMOVABLE 394
NWVOL_NAME 393
NWVOL_TOTAL_BLKS 393
ON_CHIP_APIC 399
ON_CHIP_FPU 400
OS_INST_DATE 395
OS_MAJOR_VERS 394
OS_MINOR_VERS 395
OS_NAME 394
OS_SUB_VERS 395
OS_TYPE 394
PACKAGE_ID 387
PACKAGE_NAME 387
PACKAGE_VERS 387
PAGE_ATTR_TABLE 400
PAGE_GLOBAL_ENABLE 399
PAGE_SIZE 372
PAGE_SIZE_EXT 398
PAGE_SIZE_EXT36 400
PARENT_ADDR 410
PARTITION_TYPE 395
PATH (installierte Datei) 377
PATH (nicht abgeglichen) 389, 391
PCI_DEV_NAME 402
PERM_MAC_ADDR 388
PHYSICAL_ADDR_EXT 399
PHYSICAL_FREE_KB 372
PHYSICAL_SIZE_KB 396
PHYSICAL_TOTAL_KB 372
PORT_NAME 403
PORT_NUM 410
PRFL_ACTION 413
PRINTER_IS_LOCAL 403
PRINTER_LOCATION 403
PRINTER_MODEL 403
PRINTER_NAME 403
PROCESSOR_MODEL (PC-Prozessor) 398
PROCESSOR_MODEL (Prozessor) 404
PROCESSOR_SPEED (PC-Prozessor) 398

Spaltenname, Beschreibung *(Forts.)*
PROCESSOR_SPEED (Prozessor) 404
PRODUCT 411
PRODUCT_ID 411
PUBLISHER 387
RECORD_ACTION 413
RECORD_TIME 371
SER_NUM (CD-ROM) 371
SER_NUM (Festplatte) 375
SER_NUM (PC-Prozessor) 401
SER_NUM (Prozessor) 404
SER_NUM (Speichergerät) 405
SER_NUM (USB) 410
SER_NUM_ENABLED 400
SIMD_EXT_SUPP 400
STORAGE_TYPE (CD-ROM) 371
STORAGE_TYPE (Diskette) 374
STORAGE_TYPE (Festplatte) 375
STORAGE_TYPE (Speichergerät) 405
SWARE_DESC 379
SWARE_DESC (CRC32-Vergleich) 406
SWARE_DESC (MD5-Vergleich) 408
SWARE_DESC (schneller Vergleich) 409
SWARE_DESC (Softwarevergleich) 385
SWARE_NAME 379
SWARE_NAME (CRC32-Vergleich) 406
SWARE_NAME (MD5-Vergleich) 408
SWARE_NAME (schneller Vergleich) 409
SWARE_NAME (Softwarevergleich) 385
SWARE_SIZE 379
SWARE_SIZE (CRC32-Vergleich) 406
SWARE_SIZE (MD5-Vergleich) 408
SWARE_SIZE (schneller Vergleich) 409
SWARE_SIZE (Softwarevergleich) 385
SWARE_VERS 379
SWARE_VERS (CRC32-Vergleich) 406
SWARE_VERS (MD5-Vergleich) 408
SWARE_VERS (schneller Vergleich) 409
SWARE_VERS (Softwarevergleich) 385
TIME_STAMP_COUNTER 398
TME_OBJECT_ID 370
TME_OBJECT_LABEL 370
TOTAL_PAGES 372
USB_VERS 410
USER_NAME 397
VENDOR_ID 410
Spaltenname, Beschreibung *(Forts.)*
VID_BIOS_RELDATE 412
VID_CARD_BIOS 412
VID_CARD_MODEL 411
VID_CHIP_TYPE 412
VID_COLORS 412
VID_DAC_TYPE 412
VID_HORIZNTL_RES 412
VID_MEM 412
VID_VERTICAL_RES 412
VIRT_FREE_KB 372
VIRT_MODE_EXT 398
VIRT_TOTAL_KB 372
WORKGROUP_NAME 397
Standortunabhängige Endpunkte 14
Statusinformationen
Benachrichtigung über Suchlauf 101
Definition 100
Host angeben 327
Kollektorstatus anzeigen 115, 262
konfigurieren 100, 117, 259, 321
Protokolldatei angeben 326
Sendezeitpunkt angeben 328
Sendeziel angeben 327
Suchlaufstatus anzeigen 274
Statuskollektor
Definition 5
Funktionsweise 11
konfigurieren 100
Optionen anzeigen 119, 278, 280
Optionen konfigurieren 258, 259, 321
STORAGE_DEV, Tabelle 456
STORAGE_DEV_QUERY 432
STORAGE_DEV_VIEW 405
STORAGE_TYPE (CD-ROM), Beschreibung 371
STORAGE_TYPE (Diskette), Beschreibung 374
STORAGE_TYPE (Festplatte), Beschreibung 375
STORAGE_TYPE (Speichergerät), Beschreibung 405
SUB_128MB_SUBSCRIPTIONS 437
subscription_query.sh, Skript 87
Subskribenten
Definition 6

Subskribenten *(Forts.)*
über Abfragen ermitteln 225
Suchfunktion
DMI 185
Tivoli Hardware- 184
Suchläufe
Aktualisierung anhand von Änderungen 173, 198, 280
Benachrichtigungsintervall einstellen 202, 266
benutzerdefinierte MIF-Dateien abrufen 189, 331, 348
benutzerdefinierte Skripts angeben 189, 331, 348
Dateien abgeben 182, 331
Dateien angeben 348
Daten an Konfigurations-Repository senden 175, 343, 358
Datenoptionen festlegen 173, 329
durch aktuelle Ergebnisse ersetzen 173, 280
Endpunkt-initiiert 204, 271
Erinnerungsnachrichten an Endpunkte für standortunabhängigen Zugriff 267
Erstsuchlauf 198
Fehlerbehebung 527
für BIOS-Daten 161
für Software anpassen 174, 183, 342, 358
Hardware anpassen 183, 189, 337, 354
Hardware-Komponenten auswählen 185, 189, 337, 354
Header 180, 342
nach grundlegenden Informationen 181, 342, 358
nur nach .exe-Dateien 182
nur nach EXE-Dateien 343, 359
PC-Hardware 183, 337
Prioritäten einstellen 199, 265
Prioritäten festlegen 200
Softwaresignatur 179, 342, 358
Status anzeigen 117, 306
Termin für Verteilung festlegen 201, 266
UNIX-Hardware 188, 354
Unterstützung von Endpunkten für standortunabhängigen Zugriff 268
verbindliche Verteilungen an Endpunkte für standortunabhängigen Zugriff 266
Suchläufe *(Forts.)*
Verteilungen an Endpunkte für standortunabhängigen Zugriff ausblenden 268, 327
Verteilungsoptionen festlegen 173, 326
Verzeichnisse angeben 183, 331, 348
WoL-Unterstützung 269, 329
Zeitlimit für Endpunktantwort festlegen 202, 266
Zeitlimit für Profilverteilung festlegen 202, 266
Zeitlimit für Suchlaufversuche festlegen 326
Suchlaufverfahren 200, 204
super, Berechtigungsklasse 236
SWARE_DESC, Beschreibung 379
SWARE_DESC (CRC32-Vergleich), Beschreibung 406
SWARE_DESC (MD5-Vergleich), Beschreibung 408
SWARE_DESC (schneller Vergleich), Beschreibung 409
SWARE_DESC (Softwarevergleich), Beschreibung 385
SWARE_MATCH_CRC32 406
SWARE_MATCH_MD5 407
SWARE_MATCH_QUICK 409
SWARE_NAME, Beschreibung 379
SWARE_NAME (CRC32-Vergleich), Beschreibung 406
SWARE_NAME (MD5-Vergleich), Beschreibung 408
SWARE_NAME (schneller Vergleich), Beschreibung 409
SWARE_NAME (Softwarevergleich), Beschreibung 385
SWARE_SIG, Tabelle 457
SWARE_SIZE, Beschreibung 379
SWARE_SIZE (CRC32-Vergleich), Beschreibung 406
SWARE_SIZE (MD5-Vergleich), Beschreibung 408
SWARE_SIZE (schneller Vergleich), Beschreibung 409
SWARE_SIZE (Softwarevergleich), Beschreibung 385
SWARE_VERS, Beschreibung 379

SWARE_VERS (CRC32-Vergleich), Beschreibung 406
SWARE_VERS (MD5-Vergleich), Beschreibung 408
SWARE_VERS (schneller Vergleich), Beschreibung 409
SWARE_VERS (Softwarevergleich), Beschreibung 385
SWSIGS.INI 86
Sybase
  Konfiguration 37
  Konfigurations-Repository erstellen 81, 85
  RIM-Host 37
Syntax, für Befehle 240

T

Tabellen
  benennen 136
  Betriebsdaten 366
  dem Konfigurations-Repository hinzufügen 141
  für Benutzerdaten erstellen 135, 137
  für DB2- und Informix-Datenbanken 136
  für DB2- und Informix-Datenbanksysteme 138
  mehrstufige Löschoperation 142
  Protokoll 367
  Standardwert 439, 461
Tabellen, Konfigurations-Repository
  COMPUTER 440
  COMPUTER_SYS_MEM 441
  FILE_DESC 441
  FILE_FILTER 442
  FILE_PATH 442
  H_COMPUTER 460
  H_COMPUTER_SYS_MEM 460
  H_INST_HEADER_INFO 460
  H_INST_MOUSE 460
  H_INST_NATIVE_SWARE 460
  H_INST_PARTITION 460
  H_INST_PRINTER 460
  H_INST_PROCESSOR 460
  H_INST_USB_DEV 460
Tabellen, Konfigurations-Repository *(Forts.)*
  H_INST_VID_CARD 461
  H_IP_ADDR 461
  H_IPX_ADDR 461
  H_MATCHED_SWARE 461
  H_NET_ADAPTER 461
  H_NW_SERVER 461
  H_NW_VOLS 461
  H_PC_SYS_PARAMS 461
  H_PCI_DEV 461
  H_STORAGE_DEV 461
  H_UNIX_SYS_PARAMS 461
  H_UNMATCHED_FILES 461
  HDISK 443
  HEADER_INFO 443
  INST_HEADER_INFO 444
  INST_MOUSE 444
  INST_NATIV_SWARE 445
  INST_PARTITION 445
  INST_PRINTER 446
  INST_PROCESSOR 446
  INST_USB_DEV 447
  INST_VID_CARD 447
  IP_ADDR 448
  IPX_ADDR 449
  MATCHED_SWARE 449
  MOUSE 449
  NATIV_SWARE 450
  NET_ADAPTER 451
  NW_SERVER 451
  PC_SYS_PARAMS 453
  PCI_DEV 454
  PRINTER 454
  PROCESSOR 454
  STORAGE_DEV 456
  SWARE_SIG 457
  UNIX_SYS_PARAMS 457
  UNMATCHED_FILES 458
  USB_DEV 459
  VID_CARD 459
tail 522, 523, 524, 525
TAPEDRV_QUERY 432
TEC
  mit Tivoli Inventory verwenden 60
  Statusinformationen senden 101, 118, 278, 327

Textspezifische Konventionen xxi
Threads
für Inventory Data Handler konfigurieren 101
für Kollektoren konfigurieren 98, 113
konfigurieren für Kollektoren 250, 251
TIME_STAMP_COUNTER, Beschreibung 398
Tivoli-Arbeitsoberfläche 25
Tivoli Enterprise Console 60
Tivoli Hardware-Suchfunktion 184
Tivoli Hardware-Suchfunktion, Dialogfenster 185, 189
Tivoli Inventory
Befehle 243
deinstallieren 57
Einsatzbereich 2
Funktionen 12
Funktionsweise 6
Hinweisgruppe 198
Koexistenz mit früheren Versionen 21
Komponenten 3
Sicherheit 17
UserLink 3
Web-basierte Schnittstelle 3
zusammen mit SCS verwenden 8
Tivoli Inventory deinstallieren 57
Tivoli Inventory Gateway installieren 55
Tivoli Inventory-Web-Seiten aufrufen 146
Tivoli Management Framework, Beschreibung 1
Tivoli-Scheduler 110, 113
TME 10 Web Access, Seite 144
TME_OBJECT_ID, Beschreibung 370
TME_OBJECT_LABEL, Beschreibung 370
tmersrvd, Benutzereintrag 249
TMR
Abfragen in miteinander verbundenen TMRs 88
Anzahl der Inventory Data Handler 259
Hinweise zu miteinander verbundenen TMRs 119
Ressourcen derselben Art mit demselben Namen in einer TMR 242
RIM-Host für miteinander verbundene TMRs 32
TMR-Berechtigungsklassen festlegen, Dialog 131
TMR-Verbund 32, 88, 119, 233
tnsnames.ora, Datei 37, 49
TOTAL_PAGES, Beschreibung 372

U

Übertragungsblock 98, 113, 250
Umbenennen, Profile 196
umfassende Verwaltungsumgebung 16
UNIX-Hardwaresuchläufe 188, 354
UNIX_SYS_PARAMS, Tabelle 457
UNIX_SYS_QUERY 433
UNMATCHED_FILES, Tabelle 458
USB_DEV, Tabelle 459
USB_DEV_QUERY 433
USB_DEV_VIEW 410
USB_VERS, Beschreibung 410
user, Berechtigungsklasse 236
USER_NAME, Beschreibung 397
useradd.mif, Datei
Erstellen 155
Gruppen und Attribute anzeigen 150
useradd.mifm, Datei
Erstellen 127
UserLink, Benutzerzugriff erteilen 129, 135
UserLink for Tivoli, Seite 153
UserLink for Tivoli Inventory, Seite 154
userlink.htm, Datei 152

V

VENDOR_ID, Beschreibung 410
Verwaltete Ressourcen, Standardwerte festlegen 463
Verwaltung durch Subskription 6
Verwaltungssystem für relationale Datenbanken 3
Verwendete Konventionen xxi

Verzeichnis
$BINDIR/TME/INVENTORY 187
$DBDIR 521
$DBDIR/inventory/mc 520
$DBDIR/inventory/stat_dir 259
$DBDIR/mcollect 97, 249, 520, 525
$DBDIR/mcollect/depot 527
Verzeichnis $DBDIR 521
Verzeichnis $DBDIR/inventory/mc 520
Verzeichnis $DBDIR/mcollect 97, 520, 525
Verzeichnisse, Regeln für deren Angabe 332, 349
VID_BIOS_RELDATE, Beschreibung 412
VID_CARD, Tabelle 459
VID_CARD_BIOS, Beschreibung 412
VID_CARD_MODEL, Beschreibung 411
VID_CARD_QUERY 434
VID_CARD_VIEW 411
VID_CHIP_TYPE, Beschreibung 412
VID_COLORS, Beschreibung 412
VID_DAC_TYPE, Beschreibung 412
VID_HORIZNTL_RES, Beschreibung 412
VID_MEM, Beschreibung 412
VID_VERTICAL_RES, Beschreibung 412
VIRT_FREE_KB, Beschreibung 372
VIRT_MODE_EXT, Beschreibung 398
VIRT_TOTAL_KB, Beschreibung 372

W

wadminep 512, 513, 528
Wake-on-LAN
Beschreibung 14
WAN-Verbindungen 23, 119
Warteschlange
Ausgabe 5, 521
beendete 5
Beschreibung 4
Eingabe 4, 521
Fehler 5, 521
Informationen anzeigen 263
Informationen anzeigen zu 115
Protokolldateien 521
verzögerte 5, 521
wbkupdb 38
wcollect 23, 95, 101, 111, 115, 247, 522, 534
wcomprules 61
wcprb 61
wcrtadmin 134
wcrtinvcb 104, 256
wcrtinvdh 100, 103, 118, 258, 274, 276, 321
wcrtpol 466
wcrtprf 171, 193
wcrtquery 143
wcrtrb 61
wcrtrim 51, 105
wcstat 115, 253, 262, 522
wdel 103, 104, 195
wdistinv 204, 265
Web-Seiten aufrufen 146
wepscan 204, 271, 516
wgetinvdh 116, 274, 529
wgetinvglobal 117, 119, 277
wgetinvpcfiles 282
wgetinvpchw 285
wgetinvpcsw 288
wgetinvunixfiles 294
wgetinvunixhw 297
wgetinvunixsw 300
wgetpolm 467
wgetrim 32, 529
wgetscanstat 117, 119, 306, 527, 528
Wiederholen 251
Wiederholung 98
wimprbclass 61
WIN_2000_SUBSCRIPTIONS 437
WIN_98_SUBSCRIPTIONS 437
WIN_ALL_SUBSCRIPTIONS 437
WIN_ME_SUBSCRIPTIONS 437
WIN_NT_SUBSCRIPTIONS 438
winstall 52
winvfilter 308
winvrmnode 142, 311
winvsig 124, 125, 313
wloadrb 61
wlookup 107
wlsinst 529
wlspolm 467
wmdist 202
WMI-Suchprogramm 13

WoL (Wake-on-LAN)
aktivieren 269, 329
WORKGROUP_NAME, Beschreibung 397
wpatch 29
wputpolm 468
wqueryinv 318
wrimset 535
wrimtest 528, 535
wrimtrace 526
wrpt 95, 96, 252, 529, 531
wruninvquery 228
wrunquery 221, 224, 228
wsetadmin 134
wsetinvdh 100, 118, 259, 274, 276, 321
wsetinvglobal 118, 259, 261, 275, 278, 322, 325, 524
wsetinvpcfiles 331
wsetinvpchw 337
wsetinvpcsw 342
wsetinvunixfiles 348
wsetinvunixhw 354
wsetinvunixsw 358
wsetpr 469
wsetrim 32, 108
wsetrimpw 32, 52, 66, 70, 83
wstartesvr 61
wstopesvr 61
wuninst 57

Z

Zeitlimiteinstellungen, Dialogfenster 201
Ziele 6

GC12-2612-01